



# Inversion for textured images : unsupervised myopic deconvolution, model selection, deconvolution-segmentation

Cornelia Paula Văcar

## ► To cite this version:

Cornelia Paula Văcar. Inversion for textured images : unsupervised myopic deconvolution, model selection, deconvolution-segmentation. Image Processing [eess.IV]. Université de Bordeaux, 2014. English. NNT : 2014BORD0131 . tel-01154141

**HAL Id: tel-01154141**

**<https://theses.hal.science/tel-01154141>**

Submitted on 21 May 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE  
POUR OBTENIR LE GRADE DE  
**DOCTEUR DE**  
**L'UNIVERSITÉ DE BORDEAUX**

ÉCOLE DOCTORALE SCIENCES PHYSIQUES ET DE L'INGENIEUR  
Spécialité : automatique, productique, signal et image, ingénierie cognitive

Par Cornelia Paula VĂCAR

**Inversion for textured images :  
unsupervised myopic deconvolution,  
model selection, deconvolution-segmentation**

Sous la direction de : Jean-François GIOVANNELLI  
co-directeur : Yannick BERTHOUMIEU

Soutenue le 19 septembre 2014

Membres du jury :

Thierry COLIN,  
Nicolas DOBIGEON  
Ronan FABLET  
Florence TUPIN  
Jean-François GIOVANNELLIÉ  
Yannick BERTHOUMIEU

Professeur à l'Institut Polytechnique de Bordeaux  
Maitre de Conférences à INP-ENSEEIH  
Professeur à Telecom Bretagne  
Professeur à Telecom ParisTech  
Professeur à l'Université de Bordeaux  
Professeur à l'Institut Polytechnique de Bordeaux

Président  
Rapporteur  
Rapporteur  
Examineur  
Directeur de thèse  
Co-encadrant

## **Titre : Inversion pour image texturée : déconvolution myope non supervisée, choix de modèles, déconvolution-segmentation**

**Résumé :** Ce travail est dédié à la résolution de plusieurs problèmes de grand intérêt en traitement d'images : segmentation, choix de modèle et estimation de paramètres, pour le cas spécifique d'images texturées indirectement observées (convoluées et bruitées). Dans ce contexte, les contributions de cette thèse portent sur trois plans différents : modèle, méthode et algorithmique.

Du point de vue modélisation de la texture, un nouveau modèle non-gaussien est proposé. Ce modèle est défini dans le domaine de Fourier et consiste en un mélange de Gaussiennes avec une Densité Spectrale de Puissance paramétrique.

Du point de vue méthodologique, la contribution est triple – trois méthodes Bayésiennes pour résoudre de manière :

- optimale
- non-supervisée
- des problèmes inverses en imagerie dans le contexte d'images texturées indirectement observées, problèmes pas abordés dans la littérature jusqu'à présent.

Plus spécifiquement,

1. la première méthode réalise la déconvolution myope non-supervisée et l'estimation des paramètres de la texture,
2. la deuxième méthode est dédiée à la déconvolution non-supervisée, le choix de modèle et l'estimation des paramètres de la texture et, finalement,
3. la troisième méthode déconvolue et segmente une image composée de plusieurs régions texturées, en estimant au même temps les hyperparamètres (niveau du signal et niveau du bruit) et les paramètres de chaque texture.

La contribution sur le plan algorithmique est représentée par une nouvelle version rapide de l'algorithme Metropolis-Hastings. Cet algorithme est basé sur une loi de proposition directionnelle contenant le terme de la "direction de Newton". Ce terme permet une exploration rapide et efficace de l'espace des paramètres et, de ce fait, accélère la convergence.

**Mots clés :** problème inverse, BAYES, segmentation, hyper-paramètre, déconvolution

---

## **Title : Inversion for textured images : unsupervised myopic deconvolution, model selection, deconvolution-segmentation**

**Abstract :** This thesis is addressing a series of inverse problems of major importance in the field of image processing (image segmentation, model choice, parameter estimation, deconvolution) in the context of textured images. In all of the aforementioned problems the observations are indirect, i.e., the textured images are affected by a blur and by noise. The contributions of this work belong to three main classes: modeling, methodological and algorithmic.

From the modeling standpoint, the contribution consists in the development of a new non-Gaussian model for textures. The Fourier coefficients of the textured images are modeled by a Scale Mixture of Gaussians Random Field. The Power Spectral Density of the texture has a parametric form, driven by a set of parameters that encode the texture characteristics.

The methodological contribution is threefold and consists in solving three image processing problems that have not been tackled so far in the context of indirect observations of textured images. All the proposed methods are Bayesian and are based on the exploiting the information encoded in the a posteriori law. The first method that is proposed is devoted to the myopic deconvolution of a textured image and the estimation of its parameters. The second method achieves joint model selection and model parameters estimation from an indirect observation of a textured image. Finally, the third method addresses the problem of joint deconvolution and segmentation of an image composed of several textured regions, while estimating at the same time the parameters of each constituent texture.

Last, but not least, the algorithmic contribution is represented by the development of

a new efficient version of the Metropolis Hastings algorithm, with a directional component of the proposal function based on the "Newton direction" and the Fisher information matrix. This particular directional component allows for an efficient exploration of the parameter space and, consequently, increases the convergence speed of the algorithm.

To summarize, this work presents a series of methods to solve three image processing problems in the context of blurry and noisy textured images. Moreover, we present two connected contributions, one regarding the texture models and one meant to enhance the performances of the samplers employed for all of the three methods.

**Keywords :** Inverse problems, BAYES, segmentation, Hyper-parameters, deconvolution

---

## Unité de recherche

Laboratoire IMS, CNRS UMR-5218, Groupe Signal et Image

*Părinților mei, Paula și Cornel*

*Surorii mele, Laura*



# Acknowledgements

Thanking all the people that have contributed directly or indirectly to the smooth conduct of my work may prove to be a hard task. I hope however that I will not forget to mention any of the people that have guided, inspired, supported or encouraged me throughout this endeavour.

Perhaps the best way to start is by thanking the two people who made this thesis possible : Jean-François and Yannick. First of all, Jean-François for believing in me and for deciding without hesitation that you wanted to work with me on this topic...enough to wait for 1 year for me to obtain my M.Sc. degree. Also, thank you for always being the voice of reason, the one who always considered all the options and put them in balance and for sometimes tempering my impulsiveness. But, most of all, I am grateful for your guidance and your relentlessness in sharing your knowledge and your ideas. Thank you Yannick for always being so contagiously energetic, with a smile on your face and a funny word to say in any circumstance. And thank you both for taking interest in my long term career projects and supporting me in pursuing them.

I warmly thank my colleagues in the Signal and Image Processing Team at IMS for the good moments we have shared in the 5 years I have been around. I have seen people come and go, but I always took pleasure in the company of Valerie, Céline, Jean-Pierre, Christian, Guillaume F., Pascal and Marc. Thank you Audrey for always being so positive and keeping us up to date with who's dating who and thank you Lionel for not making public the secrets you have on all of us.

I wish to also thank my office colleagues Ioana, Olivier and Guillaume R. for accepting me among them, even if I wasn't a "teledetectionist". Special thanks to Olivier for helping me clean up the mess I often made by spilling my coffee and for not being afraid of being seated so close to me. And to Julien who made sure our lunch breaks were never too short and who took his coffee break with us without ever having coffee. I also thank Guillaume R. for only throwing my boots across the hallway and never off the window.

I am also very happy for having survived several troubling events in the team, such as the kidnapping of Julien's cactus by the "El Cartel" (safely returned to its owner in the end), the transformation of scientific posters into nudity revealing material, dangerous chair races and many more...

I would also like to mention the support of a few very special friends who have been close to me throughout this time. Among them, Maria, Denisia, Roxana, Irina, Serena, Vincent, thank you for listening and encouraging me.

I am particularly grateful to my parents for their unconditional love and support and for all the energy they have invested in me. My sister has also been extremely supportive and motivated to stay in contact with my work, so she read all my scientific papers (without having any image processing background). I am so fortunate to have you !!!!

A very special thought goes to Guillaume, for all the patience and helping me keep my balance in my moments of doubt. Thank you for being there for me, listening and encouraging me.





# Contents

Résumé . . . . .	vii
Résumé en français du contenu du mémoire . . . . .	ix
Contexte et formulation du problème . . . . .	ix
Conclusion et perspectives . . . . .	xv
Abstract . . . . .	xix
<b>1 Introduction</b>	<b>1</b>
1.1 Context and Problem statement . . . . .	1
1.2 Contributions . . . . .	4
1.3 Manuscript Structure . . . . .	4
<b>2 Inverse Problems, Bayesian Framework and Stochastic Sampling</b>	<b>7</b>
2.1 Inverse Problems in Image Processing . . . . .	7
2.1.1 Indirect Data . . . . .	8
2.1.2 Full Inversion Problem . . . . .	10
2.2 Bayesian Approach – Prior, Joint and Posterior Laws . . . . .	11
2.2.1 Uninformative Priors. Conjugacy . . . . .	12
2.2.2 Estimation . . . . .	12
2.3 Addressed Problems . . . . .	13
2.4 Efficient Metropolis-Hastings Samplers . . . . .	13
2.4.1 Independent . . . . .	15
2.4.2 Standard Random-Walk . . . . .	16
2.4.3 Langevin adapted Random-Walk . . . . .	16
2.4.4 Hessian adapted Random-Walk . . . . .	17
2.4.5 Fisher adapted Random-Walk . . . . .	18
2.5 Conclusion . . . . .	20
<b>3 Texture Modeling</b>	<b>21</b>
3.1 Introduction and State of the art . . . . .	21
3.2 Texture Modeling by Random Fields . . . . .	25
3.2.1 Gaussian Model . . . . .	27
3.2.2 Non-Gaussian Model . . . . .	32
3.3 Conclusion and Perspectives . . . . .	35
<b>4 Unsupervised Myopic Deconvolution of a Textured Image</b>	<b>37</b>
4.1 Problem Statement . . . . .	37

4.1.1	Deconvolution . . . . .	39
4.1.2	Data, image and TF models . . . . .	41
4.2	Myopic deconvolution for textured images . . . . .	42
4.2.1	Information and qualitative estimation performance analysis . . . . .	42
4.2.2	Bayesian setting: priors, posterior and conditional posteriors . . . . .	46
4.3	Results . . . . .	48
4.4	Conclusion and Perspectives . . . . .	51
<b>5</b>	<b>Model Choice for the Law and the PSD of a Textured Image</b>	<b>53</b>
5.1	Model Choice - State of the art . . . . .	53
5.2	Problem Statement . . . . .	54
5.2.1	Texture coefficients law and PSD models . . . . .	54
5.2.2	Probabilistic model choice . . . . .	56
5.2.3	Joint law and priors for the model, image and parameters . . . . .	57
5.3	Evidence calculation . . . . .	58
5.3.1	Evidence approximations based on posterior samples . . . . .	58
5.3.2	Posterior sampling . . . . .	60
5.3.3	Gibbs within-model posterior sampling . . . . .	61
5.3.4	Implementation issues . . . . .	63
5.4	Experimental Results . . . . .	63
5.4.1	RWMH vs FMH . . . . .	65
5.4.2	HMA and LMA . . . . .	66
5.4.3	CEAPS performances . . . . .	66
5.4.4	Visual reconstructions . . . . .	68
5.5	Conclusion and Perspectives . . . . .	70
<b>6</b>	<b>Deconvolution Segmentation for Textured Images</b>	<b>73</b>
6.1	State of the art . . . . .	73
6.2	Problem Statement . . . . .	75
6.3	Bayesian Formulation . . . . .	77
6.3.1	Estimators . . . . .	78
6.3.2	Computing the Estimators – <i>A Posteriori</i> Conditionals . . . . .	79
6.4	Sampling Aspects . . . . .	80
6.4.1	Sampling the Labels . . . . .	80
6.4.2	Sampling the Image . . . . .	82
6.5	Results . . . . .	87
6.5.1	Evaluation of the exact method . . . . .	89
6.5.2	Influence of the $\beta$ parameter . . . . .	96

<b>Contents</b>	<b>v</b>
6.5.3 Influence of the approximations . . . . .	97
6.6 Conclusion and Perspectives . . . . .	98
<b>7 Conclusion and Perspectives</b>	<b>101</b>
<b>A Fisher information for indirectly observed GRF textures</b>	<b>105</b>
<b>B Optimal Bayesian Estimation</b>	<b>107</b>
B.1 Posterior Mean . . . . .	107
B.2 Evidence based Classifier . . . . .	108
<b>C Potts Model</b>	<b>111</b>
<b>D Truncation Matrices</b>	<b>113</b>



# Résumé

Ce travail est dédié à la résolution de plusieurs problèmes de grand intérêt en traitement d'images : segmentation, choix de modèle et estimation de paramètres, pour le cas spécifique d'images texturées indirectement observées (convoluées et bruitées). Dans ce contexte, les contributions de cette thèse portent sur trois plans différents : modèle, méthode et algorithmique.

Du point de vue modélisation de la texture, un nouveau modèle non-gaussien est proposé. Ce modèle est défini dans le domaine de Fourier et consiste en un mélange de Gaussiennes avec une Densité Spectrale de Puissance paramétrique.

Du point de vue méthodologique, la contribution est triple – trois méthodes Bayésiennes pour résoudre de manière :

- optimale
- non-supervisée
- des problèmes inverses en imagerie dans le contexte d'images texturées indirectement observées, problèmes pas abordés dans la littérature jusqu'à présent.

Plus spécifiquement,

1. la première méthode réalise la déconvolution myope non-supervisée et l'estimation des paramètres de la texture,
2. la deuxième méthode est dédiée à la déconvolution non-supervisée, le choix de modèle et l'estimation des paramètres de la texture et, finalement,
3. la troisième méthode déconvolue et segmente une image composée de plusieurs régions texturées, en estimant au même temps les hyperparamètres (niveau du signal et niveau du bruit) et les paramètres de chaque texture.

La contribution sur le plan algorithmique est représentée par une nouvelle version rapide de l'algorithme Metropolis-Hastings. Cet algorithme est basé sur une loi de proposition directionnelle contenant le terme de la "direction de Newton". Ce terme permet une exploration rapide et efficace de l'espace des paramètres et, de ce fait, accélère la convergence.



# Résumé en français du contenu du mémoire

## Contexte et formulation du problème

L'énorme quantité de données numériques produites actuellement peuvent être utilisées dans des domaines très variés, néanmoins, leur très grand volume rend impossible leur traitement par des spécialistes. Par conséquent, il y a un besoin réel de méthodes de traitement automatique de données pour le contrôle de la qualité en industrie, l'aide au diagnostic dans des applications biomédicales, systèmes de vidéo-surveillance et nombreuses autres applications. Dans ce contexte, le traitement d'images est devenu une axe centrale de recherche scientifique.

Par ailleurs, la majorité des images naturelles comportent de la texture. En général, la texture contient une grande quantité d'information concernant la scène. Pour cette raison, dans le domaine du traitement d'images, les problématiques liées à la texture ont une importance particulière. Par conséquent, des nombreuses méthodes de modélisation, synthèse et analyse de textures ont été développées. Pour donner juste quelques exemples d'applications, ces méthodes sont utilisées dans des domaines comme :

- **l'industrie** : pour l'analyse et l'inspection de la structure de matériaux, la détection des défauts, l'analyse des modèles des tissus, le contrôle de la production ;
- **imagerie biomédicale** : segmentation d'organes ou de vaisseaux, l'analyse structurale des tissus afin de détecter des anomalies pouvant indiquer la présence de maladies, la détection de tumeurs, l'estimation du mouvement pour étudier le cycle cardiaque ou respiratoire ou la structure de différentes structures anatomiques ;
- **traitement d'images satellite** : segmentation d'images pour des raisons cartographiques, classification de régions (urbaine, rurale, exploitation agricole, environnement naturel), classification de cultures (vigne, forêt), estimation et prédiction de la productivité d'une culture, évaluation de dommages (tempête, incendie) ;
- **investigations géologiques** : caractérisation des structures géologiques, à partir de différentes techniques d'imagerie.

Les applications dédiées à la texture consistent en la définition d'un modèle pour la texture et l'apprentissage des caractéristiques de ce modèle à partir d'un certain nombre de textures de référence. Cette information peut être utilisée pour réaliser deux tâches principales :

- **analyse de la texture** : *i.e.*, l'extraction de l'information pertinente décrivant les principales caractéristiques de la texture. Cette analyse peut être basée sur l'interaction entre les pixels dans différents espaces de représentation (spatial, ondelettes), sur la distribution des fréquences dans le domaine de Fourier, sur l'orientation des

vecteurs ou tenseurs, sur la saillance, sur l'entropie, *etc.*... Plus spécifiquement, ça revient à trouver des mesures qui décrivent l'image, appelées **attributs**, qui permettent d'identifier les textures similaires et de distinguer les textures différentes.

- **synthèse de la texture** : en utilisant modèle déterminé à partir des textures de référence, des nouvelles images ayant des caractéristiques similaires et appartenant à la même famille de textures peuvent être générées.

Un aspect très important est que pas tous les modèles sont génératifs, par exemple, les méthodes d'analyse de textures basées sur des attributs peuvent être utilisées seulement pour classer ou comparer des textures et pas pour synthétiser une nouvelle texture.

Ce travail représente un étude sur des divers problèmes inverses en traitement d'images, appliqués au cas spécial de la texture. Les problèmes inverses qui sont traités sont la **déconvolution**, le **débruitage**, la **segmentation**, l'**estimation des paramètres** et le **choix de modèle**. Plus spécifiquement, trois méthodes ont été développées :

- I. une méthode de déconvolution myope, qui estime les hyperparamètres (est non-supervisée), les paramètres de l'instrument et les paramètres de la texture,
- II. une méthode pour le choix de modèle de la forme de la matrice de covariance pour des images texturées indirectement observées, *i.e.*, floutées et bruitées,
- III. une méthode pour la déconvolution-segmentation d'images texturées, qui estime au même temps le niveau du bruit et les paramètres des textures.

Tous ces problèmes sont très mal-posés, c'est-à-dire que les données observées ne suffisent pas pour obtenir une solution unique. Pour cela, de l'information supplémentaire est nécessaire. Une manière simple et naturelle est la formulation Bayésienne où l'information supplémentaire est incluse sous la forme des lois *a priori*. En fait, toutes les méthodes présentées dans ce manuscrit sont basées sur une formulation Bayésienne, où l'information contenue par les données est exploitée via le terme d'adéquation aux données (la vraisemblance), tandis que l'information *a priori* constitue le terme de régularisation.

Dans le cadre des approches Bayésiennes, toute l'information est encodée par la loi *a posteriori*. Des estimateurs sont définis à partir de cette loi, leur rôle étant d'exploiter l'information afin de trouver une solution du problème. En fonction de l'estimateur qui est utilisé, la solution peut avoir certaines propriétés d'optimalité. Par exemple, dans ce travail, la Moyenne *A Posteriori* est utilisée très souvent, sachant que cet estimateur minimise l'erreur quadratique moyenne. Pour calculer les estimés, la Moyenne *A Posteriori* utilise des échantillons de la loi *a posteriori*. Néanmoins, comme dans le cas de la plupart des problèmes non-triviales, cette loi a une forme compliquée et ne peut être échantillonnée directement. Par conséquent, des méthodes de Monte Carlo par Chaîne de Markov doivent être employées, plus spécifiquement l'échantillonnage de Gibbs et, pour les lois conditionnelle non-standard, l'algorithme de Metropolis-Hastings.

Les problèmes d'inversion mentionnées précédemment sont traitées dans le contexte d'images texturées. De plus, le focus est mis sur une classe particulière de textures, modélisée par des champs aléatoires centrés et stationnaires, avec une Densité Spectrale de Puissance paramétrique. Le modèle de texture basé sur de champs aléatoires gaussiens,



qui représente le premier modèle exploité dans ce travail, possède des caractéristiques très avantageuses qui permettent son intégration dans le paradigme d'inversion. D'un autre côté, même si ce modèle est relativement versatile, il souffre de quelques limitations :

- il est basé seulement sur les statistiques d'ordre 2,
- les coefficients de Fourier de la texture sont indépendants.

L'extension de ce modèle à une loi non-gaussienne n'est pas une tâche facile, car le but est d'augmenter sa capacité de représentation, tout en gardant sa complexité à un niveau qui permet son intégration dans notre formalisme de traitement d'images. Dès lors, il y a un compromis à faire entre la versatilité et la complexité du modèle, *i.e.*, afin de pouvoir gérer facilement le modèle, certaines limitations vont être gardés.

## Contributions

Les principales contributions présentées dans ce manuscrit reflètent la dualité de ce travail. D'abord, le focus sur les images texturées a ouvert la question sur le besoin d'un modèle de texture qui est à la fois facilement gérable et qui a des capacités de représentation satisfaisantes. Ensuite, en utilisant ce modèle, de problèmes compliqués en traitement d'images ont été abordés dans le contexte des images texturées. Finalement, la résolution de ces problèmes de manière optimale a nécessité l'utilisation de l'échantillonnage. En général, l'échantillonnage n'est pas un outil très souvent utilisé en traitement d'images, à cause de la grande dimension des problèmes. Néanmoins, grâce aux propriétés du modèle de texture proposé, l'échantillonnage n'est pas trop coûteux dans ce cas. De plus, pour améliorer les performances de ces méthodes, nous avons développé un algorithme efficace d'échantillonnage pour des lois cible compliquées.

Ces contributions sont de nature assez différente. **La première contribution est liée à la modélisation :**

**Mo1.** Proposition d'un modèle non-gaussien de texture, basé sur les moments d'ordre 2, et son intégration dans un problème compliqué de traitement d'images,

**les prochains trois contributions sont de nature méthodologique** – développement et mise en œuvre de méthodes pour :

**Me1.** la déconvolution myope, avec estimation des hyperparamètres, paramètres instrument et paramètres de la texture,

**Me2.** le choix du modèle de la Densité Spectrale de Puissance d'une image texturée, à partir d'observations indirectes,

**Me3.** la déconvolution-segmentation d'images texturées, avec estimation des hyperparamètres et paramètres de la texture,

tandis que **la dernière contribution est algorithmique :**

- A1. Développement d'une version efficace de l'algorithme d'échantillonnage de Metropolis-Hastings, basé sur la matrice d'information de Fisher, et son intégration dans nos algorithmes.

## Contenu de la thèse

### Chapitre 2 : Problèmes inverses, cadre bayésien et échantillonnage stochastique

Ce chapitre est dédié à l'introduction des notions qui seront récurrentes pendant ce travail, par exemple l'idée d'observations indirectes et les divers problèmes inverses qui peuvent être abordés dans ce contexte. Le système d'observation choisi introduit un flou et un bruit blanc gaussien additif. De plus, le flou est considéré comme ayant une forme paramétrique avec des paramètres  $\eta$  et la loi du bruit est pilotée par le paramètre  $\gamma_n$ . Sachant que ces paramètres sont inconnus, ce travail va viser à résoudre le problème de la déconvolution dans un cas non-supervisé (estimer  $\gamma_n$ ) et myope (estimer  $\eta$ ).

Le problème complet qui peut se poser dans ce cas pour des images texturées est décrit dans la suite. Celui-ci combine les aspects non-supervisés et myopes, associés aux observations indirectes, avec la sélection du modèle de texture et la segmentation d'images texturées. Ce problème est fortement mal posé à cause du déficit d'information (le nombre d'inconnues est nettement plus grand que celui des observations). Ce problème inverse sera traité dans un cadre bayésien, où l'information contenue par les données est extraite via la loi de vraisemblance et la régularisation est réalisée en imposant des modèles pour les inconnues et des lois *a priori* sur les paramètres de ces lois. La loi *a posteriori* peut être exprimée à partir de ces ingrédients et des estimateurs pour les inconnues seront formulés. Vu que ces estimateurs ne peuvent pas être calculés de manière analytique, ils seront calculés numériquement en utilisant un échantillonneur de Gibbs, qui consiste à échantillonner de manière itérative les lois *a posteriori* conditionnelles.

Dans ce contexte, ce chapitre introduit la contribution algorithmique, un échantillonneur efficace du type Metropolis-Hastings pour des lois cible compliquées. Cet échantillonneur est basé sur une proposition directionnelle similaire à la direction de Newton. La spécificité de cette proposition est que la matrice Hessienne est remplacée par la matrice d'information de Fisher, afin d'éliminer certaines instabilités numériques. De plus, dans notre cas spécifique de loi pour les paramètres de la texture, ce principe nous permet de profiter des avantages d'une proposition directionnelle de deuxième ordre en calculant seulement les dérivées de premier ordre.

### Chapitre 3 : Modélisation de la texture

Cette thèse est focalisée sur le traitement d'images texturées. La plupart des travaux qui portent sur ce sujet sont basés sur le principe d'analyse de la texture, sous la forme de l'extraction d'attributs texturaux et la comparaison de ces attributs. L'approche choisie

dans ce travail et d'utiliser des modèles génératifs, capables au même temps d'indiquer si une image est bien décrite par ce modèle et de produire de réalisations de textures ayant certaines caractéristiques. Un modèle souple et facile à intégrer dans le cadre de notre problématique est le modèle basé sur des Champs Aléatoires Gaussiens (CAG). Ce modèle repose sur l'hypothèse que les images sont stationnaires, ce qui implique, par approximation de Whittle, une structure circulante-bloc-circulante de la matrice de covariance. Cette structure particulière implique que la matrice est diagonalisable par Transformée de Fourier (TF) et que les coefficients de Fourier de l'image sont indépendants (décorrélés et gaussiens). Grâce à cette propriété, les coefficients de Fourier de l'image peuvent être traités en parallèle, par conséquent, de manière très efficace. De plus, dans ce travail, des formes paramétriques sont employés pour la Densité Spectrale de Puissance (DSP) des images. Ces formes sont pilotées par un nombre réduit de paramètres, ce qui implique une forte compressibilité du modèle.

La gaussianité et l'indépendance des coefficient de Fourier de l'image peuvent être considérés comme des limitations assez fortes du modèle. Néanmoins, ces propriétés jouent un rôle crucial dans la souplesse et la traçabilité du modèle. Pour cette raison, l'effort d'augmenter la capacité de représentation du modèle va s'articuler autour du principe de garder, au moins en partie ces deux propriétés. Le nouveau modèle de texture proposé dans ce chapitre est basé sur l'idée de la gaussianité conditionnelle par rapport à un ensemble de variables auxiliaires de telle manière que marginalement par rapport à ces variables, la distribution ne sera plus gaussienne. Plus spécifiquement, un ensemble de variable auxiliaires sont introduites, une par pixel, et les coefficients de Fourier de la texture on une loi conditionnelle gaussienne et une loi marginale non-gaussienne. Ce modèle permet une extension des capacités de représentation du modèle (qui peut mieux décrire des texture plus stochastiques), grâce à une DSP enrichie, tout en gardant les avantages de l'indépendance et de la gaussianité conditionnelle des coefficients de Fourier.

## Chapitre 4 : Déconvolution myope et estimation des paramètres d'images texturées

Le premier chapitre dédié à la résolution de problèmes inverses dans le contexte d'images texturées indirectement observées vise l'estimation des paramètres de l'image texturée. Cette estimation sera réalisée conjointement avec une déconvolution myope (estimation des paramètres instrument) et non-supervisée (estimation du niveau du bruit). Le formalisme bayésien offre un cadre unifié où tous ces paramètres et leur inter-dépendances sont représentés. Dans cette approche, l'information contenue par les données est exploitée via la loi de vraisemblance et la régularisation de ce problème mal-posé est réalisée à travers l'utilisation de modèles pour l'image  $x$  et la réponse instrument  $H$ . L'image  $x$  est modélisée par un CAG avec DSP paramétrique ayant un profil laplacien. Ce profil est piloté par l'ensemble de paramètres  $\theta$ , contenant les deux fréquences centrales et les dispersions fréquentielles correspondantes dans l'espace des fréquences réduites. La réponse instrument est représentée par un filtre passe-bas, avec profil de Dirichlet, centré et isotrope, piloté par le paramètre de dispersion  $\eta$ .

Afin d'avoir une description complète du problème, des lois *a priori* pour les pa-

ramètres  $\gamma_n$ ,  $\theta$  et  $\eta$  seront choisies. Pour  $\gamma_n$  une loi Gamma a été employée grâce à sa forme conjuguée par rapport à la loi de vraisemblance. En ce qui concerne les  $\theta$  et  $\eta$ , à cause de la dépendance très compliquée, il n'existe pas de forme conjuguée. Par conséquent, une loi uniforme sur le domaine de définition de ces paramètres a été choisie.

A partir de tous ces ingrédients, la loi *a posteriori* peut être exprimée. L'information contenue par cette loi est exploitée en utilisant l'échantillonnage de Gibbs. Pour les lois conditionnelles non-standard des  $\theta$  et  $\eta$ , des étapes de Metropolis-Hastings ont été intégrées dans l'échantillonneur de Gibbs. En utilisant les échantillons ainsi obtenus, des estimateurs EAP sont calculés pour chaque inconnue. Comme sous-produit, cette méthode fournit aussi une estimation de l'image originale.

## Chapitre 5 : Sélection de modèle pour des images texturées indirectement observées

Ce chapitre porte sur le problème de sélection de modèle de texture pour des images floutées et bruitées. Plus exactement, les images texturées sont modélisées par des CAG ou MCAG avec des formes de PSD paramétriques et l'objectif est de sélectionner la forme de la DSP parmi un ensemble de modèles candidats. L'aspect données indirectes est traité de manière non-supervisée (le paramètre  $\gamma_n$  est estimé), mais il ne s'agit plus d'une déconvolution myope, car la réponse instrument est considérée connue.

Ce problème est abordé dans un cadre bayésien, à travers une stratégie optimale du point de vue du risque moyen de classification. Cette stratégie est basée sur le calcul de l'évidence (autrement appelées la vraisemblance marginale). Néanmoins, l'évidence a une expression très compliquée et, par conséquent, ne peut pas être calculée de manière analytique. Pour cette raison, des méthodes numériques doivent être utilisées. Parmi les différentes options, nous avons employé la moyenne harmonique et l'approximation de Laplace-Metropolis. Il est démontré expérimentalement que les deux approximations sont équivalentes et fournissent les mêmes résultats de sélection. De plus, cette application intègre l'échantillonneur efficace Fisher Metropolis-Hastings pour la loi conditionnelle compliquée des paramètres des DSPs  $\theta$ .

## Chapitre 6 : Segmentation d'images texturées indirectement observées

Le problème de la segmentation d'images texturées floutées et bruitées est traité de manière non-supervisée dans un cadre bayésien. Les images sont composées par un nombre inconnu  $R$  de régions, chacune de ces régions contenant une texture d'une certaine classe  $k$ . Le nombre de classes  $K$  est connu. Chacune des  $K$  textures est modélisée par un CAG avec DSP paramétrique ayant un profil connu, mais des paramètres inconnus.

La composition de l'image à partir des  $K$  textures est pilotée par un champs d'étiquettes cachées. Cet ensemble d'étiquettes est modélisé par un champs de Potts. La méthode est basée sur une approche probabiliste pour estimer conjointement les étiquettes, le niveau du bruit et les paramètres de chaque classe de texture. Les estimés sont obtenus en utilisant les échantillons de la loi *a posteriori* de la manière suivante : MAP marginalisé pour les

étiquettes et EAP pour le reste des paramètres.

## Conclusion et perspectives

Ce travail a présenté une gamme variée de problèmes inverses de grande importance dans le traitement d'images, appliqués dans le cas particulier d'images texturées. Dans ce contexte, nous avons proposé une série de contributions sur un plan algorithmique, modélisation et méthodologique.

La **contribution algorithmique** est représentée par le développement d'une version efficace de l'algorithme Metropolis-Hastings. La loi de proposition de ce nouvel échantillonneur est basée sur une composante inspirée de la direction de Newton, dans laquelle la matrice hessienne est remplacée par la matrice de Fisher. Par conséquent, les problèmes liés à l'inversion de la matrice hessienne sont évités et la composante directionnelle de la proposition a sûrement la direction de descente de gradient. En outre, dans le cas particulier de notre loi conditionnelle *a posteriori*, la loi de proposition s'écrit seulement à base de la première dérivée de la loi cible.

En ce qui concerne les perspectives de ce travail, cet algorithme peut être inclus dans une grande variété d'applications, dans tous les cas où des lois de probabilité compliquées doivent être échantillonnées.

\*\*\*\*\*

La **contribution liée à la modélisation** consiste dans le développement d'un modèle pour l'analyse et la synthèse de textures, basé sur un Mélange de Champs Aléatoire Gaussiens avec Densité Spectrale de Puissance paramétrique. La forme paramétrique de la Densité Spectrale de Puissance et les valeurs de ses paramètres rassemblent les caractéristiques texturales et permettent la classification de la texture ou la synthèse de nouveaux échantillons de texture ayant les mêmes attributs que l'original. Ce modèle non-gaussien est construit en utilisant un ensemble de variables auxiliaires, de telle manière que la loi de l'image soit gaussienne, conditionnellement à ces variables, et non-gaussienne marginalement. Dans ce travail, afin d'assurer des bonnes performances en terme de vitesse, les variables auxiliaires sont considérées indépendantes.

Les perspectives dans ce direction de recherche visent l'utilisation des priors corrélés pour les variables auxiliaires. Cela signifie que les variables auxiliaires ne seront plus indépendantes, résultant probablement dans l'augmentation des capacités de représentation de notre modèle de texture. Cependant, l'effet secondaire est l'inefficacité calculatoire de l'échantillonnage d'une telle texture. Une autre piste est l'utilisation de formes plus complexes pour la Densité Spectrale de Puissance et l'étude de la possibilité d'exploiter la phase des coefficients de Fourier afin de construire des modèles plus puissants.

\*\*\*\*\*

Du point de vue **méthodologique**, notre contribution est triple, car nous traitons séparément trois problèmes inverses dans le contexte d'observations indirectes d'images tex-

turées.

D’abord, nous avons développé une méthode pour **la déconvolution myope et l’estimation des paramètres pour des images texturées convoluées et bruitées**. Ces images texturées sont modélisées par des champs aléatoires gaussiens avec des Densités Spectrales de Puissance paramétriques. La fonction d’étalement du point a une forme paramétrique aussi, appartenant à la famille exponentielle. Les paramètres de ces deux champs sont inconnus et sont estimés par notre méthode.

Ce travail peut être facilement enrichi en utilisant une forme plus compliquée pour le filtre convolutif. Une autre extension pourrait être l’utilisation de notre modèle de texture non-gaussien.

\*\*\*

La deuxième contribution méthodologique est une **méthode de sélection de modèle basée sur le calcul de l’évidence dans le contexte d’observations indirectes d’images texturées**. Comme dans le cas précédent, les images texturées sont affectées par un flou et par du bruit et le but est de sélectionner le modèle et les valeurs des paramètres de la Densité Spectrale de Puissance de la texture. Dans ce cas, le filtre convolutif est connu, *i.e.*, nous ne positionnons pas dans un cadre déconvolution myope. Toutefois, le niveau du bruit n’est pas connu et doit être estimé. Les textures sont modélisées soit par des ”Scale Mixture of Gaussian Random Fields”, soit par des champs aléatoires gaussiens, et la méthode choisit le modèle qui a plus probablement généré la réalisation de la texture. La méthode est basée sur de l’échantillonnage et sa mise en œuvre inclut l’échantillonneur efficace Fisher Metropolis-Hastings.

Dans la continuité de ce travail, les dictionnaire de formes pour la Densité Spectrale de Puissance pourrait être étendu. Une autre perspective est l’estimation du filtre convolutif.

\*\*\*

Une dernière contribution méthodologique est une méthode pour la **déconvolution-segmentation d’images texturées, indirectement observées**. Comme précédemment, le filtre convolutif est connu et le niveau du bruit est estimé. Cette méthode arrive à fournir de très bons résultats pour un problème de difficulté significative. Dans le contexte où la segmentation de textures est en soi-même une tâche difficile, il n’y a pas de travaux dans la déconvolution-segmentation d’images texturées. Notre méthode utilise un modèle de Potts pour les étiquettes, avec un paramètre de température réglé à la main, et un modèle de champs aléatoire gaussien pour les textures. Pour chaque texture, le modèle de sa Densité Spectrale de Puissance est connu, tandis que ses paramètres doivent être estimés.

Hormis les défis théoriques significatives de ce problème, sa mise en œuvre nous a confrontés avec la tâche difficile d’échantillonner les étiquettes et les pixels de manière efficace, afin d’assurer un coût calculatoire acceptable. Cette contrainte nous a guidé graduellement vers cette version finale de formulation du problème, présentée dans ce manuscrit. Néanmoins, cette version est le résultat d’une série d’alternatives que nous avons explorées. Parmi ces formulations, celle que nous avons choisie est la seule qui nous a permis au même temps de représenter correctement les dépendances entre les variables et d’obtenir de bonnes performances algorithmiques.

Ce sujet peut être encore développé en intégrant l'estimation du paramètre de température. Dans ce cas, la méthode sera capable de s'adapter automatiquement à la topologie de l'image traitée. Une autre perspective serait d'estimer le filtre convolutif.

\*\*\*

Finalement, un projet très ambitieux serait de combiner tous ces problèmes pour obtenir une méthode qui, à partir d'observations convoluées et bruitées d'images texturées, arrive à estimer le filtre convolutif, segmenter l'image, sélectionner le modèle et estimer les paramètres des Densités Spectrales de Puissance de chaque texture présente dans l'image.

\*\*\*\*\*

Toutes les méthodes qui ont été développées sont basées sur des estimateurs optimaux, comme la Moyenne *A Posteriori*, pour l'estimation des paramètres, et le classifieur basé sur le calcul de l'évidence, pour le choix du modèle. Par conséquent, les méthodes elles-mêmes sont optimales du point de vue de l'erreur quadratique moyenne et du risque de classification moyen.

L'objectif principal de ce travail a été d'offrir des réponses à des questions qui n'avaient pas été encore posées ou résolues dans le contexte d'observations indirectes d'images texturées.





# Abstract

This thesis is addressing a series of inverse problems of major importance in the field of image processing (image segmentation, model choice, parameter estimation, deconvolution) in the context of textured images. In all of the aforementioned problems the observations are indirect, *i.e.*, the textured images are affected by a blur and by noise. The contributions of this work belong to three main classes: modeling, methodological and algorithmic.

From the modeling standpoint, the contribution consists in the development of a new non-Gaussian model for textures. The Fourier coefficients of the textured images are modeled by a Scale Mixture of Gaussians Random Field. The Power Spectral Density of the texture has a parametric form, driven by a set of parameters that encode the texture characteristics.

The methodological contribution is threefold and consists in solving three image processing problems that have not been tackled so far in the context of indirect observations of textured images. All the proposed methods are Bayesian and are based on the exploiting the information encoded in the *a posteriori* law. The first method that is proposed is devoted to the myopic deconvolution of a textured image and the estimation of its parameters. The second method achieves joint model selection and model parameters estimation from an indirect observation of a textured image. Finally, the third method addresses the problem of joint deconvolution and segmentation of an image composed of several textured regions, while estimating at the same time the parameters of each constituent texture.

Last, but not least, the algorithmic contribution is represented by the development of a new efficient version of the Metropolis Hastings algorithm, with a directional component of the proposal function based on the "Newton direction" and the Fisher information matrix. This particular directional component allows for an efficient exploration of the parameter space and, consequently, increases the convergence speed of the algorithm.

To summarize, this work presents a series of methods to solve three image processing problems in the context of blurry and noisy textured images. Moreover, we present two connected contributions, one regarding the texture models and one meant to enhance the performances of the samplers employed for all of the three methods.



# Introduction

---

## 1.1 Context and Problem statement

Humanity currently produces an exponentially increasing amount of information under the form of digital data, a significant part of which consists in images and videos. This huge amount of information could be exploited in various domains, nevertheless, the huge volume makes the processing by human specialists prohibitive. Consequently, while this data has allowed us to document and better understand our world, it has also given rise to a need for automatic processing to perform relevant data selection and provide preliminary results. We are thus witnessing an increasing need to automatize the processing in fields such as industrial quality control, medical diagnosis, image analysis and video surveillance systems. In this context, image processing has become a central axis of scientific research.

Moreover, most natural images contain texture, which generally encodes a great amount of information concerning the scene. For this reason, in the field of image processing, the texture related topics represent a field of significant importance. The considerable attention that has been paid to these topics has resulted in the development of a multitude of modeling, synthesis and analysis methods. To list only a few of the possible applications, let us refer to the fields of:

- **industry:** material structure analysis and inspection, flaws detection, fabric pattern analysis, production control;
- **biomedical imaging:** organ or vessel segmentation, tissue analysis for determining whether there are abnormalities that may indicate the presence of a disease, tumor detection, motion estimation to study the cardiac or the breathing cycle;
- **satellite image processing:** image segmentation for cartographic purposes, region recognition (urban, rural, crop, natural), crop recognition (vineyards, forests), crop production estimation and predictions, damage evaluation (what percentage of a crop / forest has been damaged by a storm / fire).
- **geological investigations:** characterizing the geological structures based on various scanning techniques.

The texture related applications consist in defining a model for the texture and acquiring information regarding that model based on observing a certain number of reference textures. Then, this information can be used to perform the two main tasks:

- **texture analysis:** or description, *i.e.*, the extraction of pertinent information describing the main textural characteristics. It can be based on pixel interaction in various

representation spaces (spatial, wavelet), on the frequencies distribution in the Fourier domain, on orientation vectors or tensors, on saliency, on entropy, *etc.*... More specifically, it consists in finding measures that best describe the texture, called **features**, that permit us to identify similar textures and to distinguish the textures that are different.

- **texture synthesis**: using the model that has been determined using the reference textures, new images can be generated that have similar characteristics and that belong to the same family of textures, as it is defined by the model.

An important aspect is that not all the models are generative, *i.e.*, some of them (the feature based texture analysis methods) can only be used for comparison and classification purposes and cannot be used to synthesize a new texture.

This thesis presents a study on inverse problems in image processing applied to the special case of textures. The inverse problems we are dealing with are **deconvolution**, **denoising**, **segmentation**, **parameter estimation** and **model choice**. More specifically we will present:

- I. a myopic deconvolution method, that also estimates the hyperparameters, the instrument parameters and the texture parameters,
- II. a model choice method for the form of the covariance matrix for an indirectly observed texture, *i.e.*, a blurry and noisy version of the texture,
- III. a method for the joint deconvolution-segmentation of a textured region, that also estimates the noise level and the texture parameters.

All of the previously mentioned problems are very ill-posed, meaning that the data alone does not contain sufficient information as to provide a unique solution. For this reason, supplementary information must be provided and this can be done in several manners. A simple and natural way to do this is to use a Bayesian framework and embed all the extra information in the *a priori* laws. In fact, all the methods presented in the following rely on a Bayesian formulation, where the information contained by the observations is exploited via the data adequacy term (the likelihood), while the *a priori* information forms the regularization term.

In the previously described Bayesian formulation for our problems, all the available information is embedded in the *a posteriori* law. Estimators are then defined based on this law, their role being to exploit the information in order to yield a solution for the problem. Depending on the estimator that is being used, the solution can have some optimality properties. For instance, in this work, the Minimum Mean Square Error estimator is predominant, which, as specified by its name, is the estimator that minimizes the mean square error. In order to compute the estimates, the Minimum Mean Square Error relies on samples of the *a posteriori* law. However, as it is the case in most non-trivial problems, this law has a complicated form and thus cannot be sampled directly. To this end, Monte Carlo Markov Chain methods will be employed, more specifically, Gibbs sampling and, for the parameters with non-standard conditional laws, the Metropolis-Hastings sampler.

The inversion problems previously described are applied to textured images. Moreover,

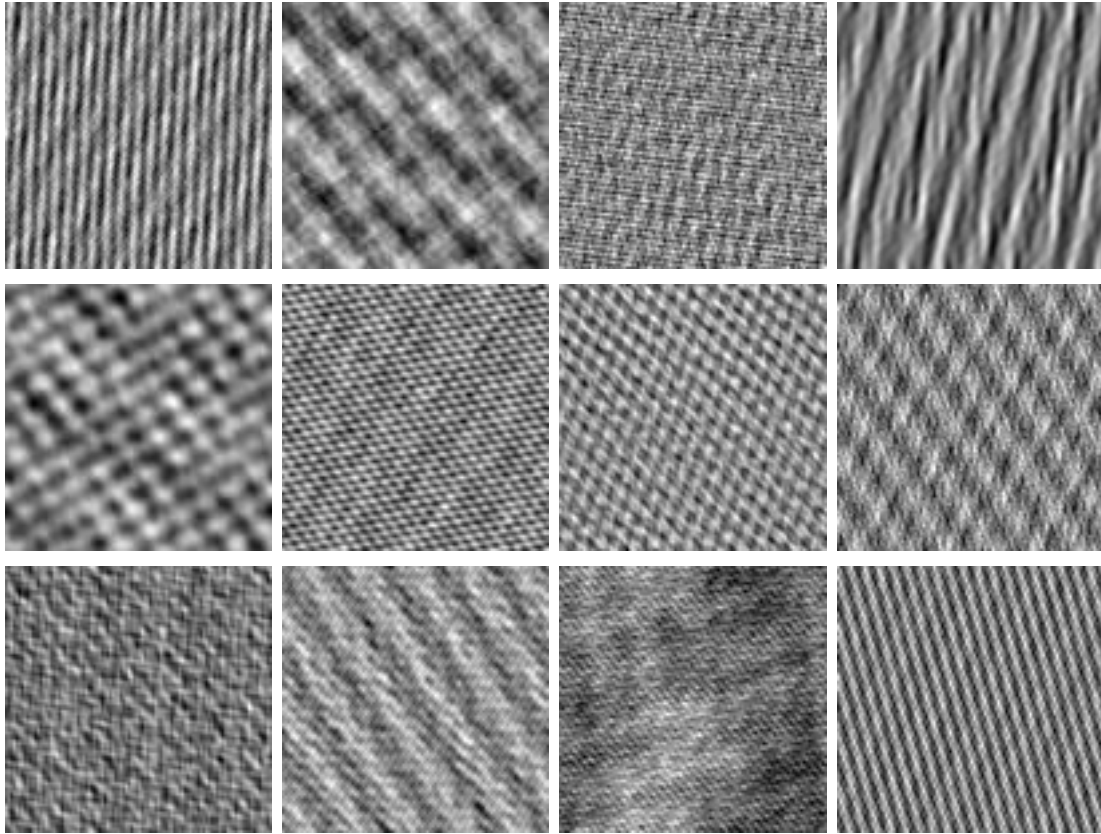


Figure 1.1: Texture realizations

we focus on a particular type of textures, modeled by stationary, zero-mean Random Fields with structured covariance, *i.e.*, a parametric form for the Power Spectral Density. The texture model based on Gaussian Random Fields, which represents the first model we have explored, presented a series of extremely advantageous features, enabling its integration in the inversion paradigm. Although this model is rather versatile and yields a large variety of textures, it suffers from several limitations:

- it is only based on the first and second order statistics,
- the texture Fourier coefficients are independent.

Extending the model to a non-Gaussian law, in a manner that increases its representation capacity, but at the same time keeps its complexity at a level that permits its integration in the larger image processing formalism, is not an easy task. Consequently, there is a compromise to be made between model versatility and complexity and this means that in order to keep the model handleable, some of the aforementioned limitations will be preserved. Figure 1.1 shows several texture realizations for a model with a parametric Power Spectral Density.

## 1.2 Contributions

The main contributions presented in this manuscript reflect the duality of our work. On the one hand, our focus on textured images has confronted us with the need for a texture model that is at the same time tractable and has strong representation capabilities. On the other hand, using on the aforementioned texture model, we were able to tackle image processing problems that are very complicated in the context of textured images. Finally, solving these problems in an optimal manner has required the use of sampling. Generally, sampling in the context of image processing is not a popular tool, due to the large dimension of the unknown. Nevertheless, due to the properties of our texture model, sampling is not prohibitively costly in our case. To further enhance the performances of our methods, we have devised an efficient sampling technique for complicated target laws.

As it can easily be noticed, our contributions are of relatively different nature. To be more exact, **the first contribution is model-related**:

**Mo1.** Development of a non-Gaussian texture model, based on the first and second order moments, and its integration in a complex image processing problem,

**the following three contributions are of methodological nature**:

**Me1.** Devising and implementing a myopic deconvolution method with joint estimation of the hyperparameters, instrument parameters and texture parameters,

**Me2.** Devising and implementing a method for model selection for the Power Spectral Density of a textured image, using an indirect observation,

**Me3.** Devising and implementing a deconvolution-segmentation method for textured images with joint estimation of the hyperparameters and texture parameters,

while **the last contribution is algorithmic**:

**A1.** Development of an efficient version of the Metropolis-Hastings sampler, based on the Fisher Information Matrix and its integration in our algorithms.

## 1.3 Manuscript Structure

This manuscript is organized as follows:

- Chapter 2 - **Inverse Problems, Bayesian Framework and Stochastic Sampling** is devoted to defining the specificities of the inverse problems encountered throughout this work. The observation model is presented and a full inverse problem is defined in this context. More specifically, based on an observed blurred and noisy textured image, we would like to determine the blurring function, the noise level, the texture model and the parameters of this model. This problem having a very large number of unknowns is addressed in a Bayesian framework. The priors for the parameters

are defined in this Chapter. Since this problem does not have an explicit solution, the Posterior Mean estimator is employed. This Chapter also provides a description of the numerical algorithms employed for sampling and the various versions of these samplers, along with our first contribution, the efficient sampler Fisher Metropolis-Hastings.

- Chapter 3 - **Texture Modeling** presents our new non-Gaussian texture model based on a Scale Mixture of Gaussian Random Fields (SMGRF). The Power Spectral Density of these Random Fields is parametric and this Chapter explains the significance of the parameters and provides priors. This represents in fact the second contribution of our work.
- Chapter 4 - **Unsupervised Myopic Deconvolution of a Textured Image** focuses on the first methodological contribution. The method presented here relies on a fully parametric formulation of the observation system and of the image model. Consequently, the unsupervised myopic deconvolution problem becomes a three-fold parameter estimation problem: observation system parameters, image model parameters and hyperparameters.
- Chapter 5 - **Model Choice for the Law and the PSD of a Textured Image** is devoted to our second methodological contribution, a method to select the Power Spectral Density model for a texture among a series of candidates, based on blurred and noisy observations.
- Chapter 6 - **Deconvolution Segmentation for Textured Images** presents the third methodological contribution: a method for textured image segmentation from blurred and noisy data. This Bayesian method assigns a prior to the labels and provides a global approach for estimating the unknowns (the labels, the original image, the noise precision).
- Chapter 7 - **Conclusion and Perspectives** lists the most important aspects that have been clarified throughout this thesis, the most interesting remarks and the main questions that remain unanswered concerning the addressed problems.

In support of the technical information comprised in the aforementioned Chapters, we have provided detailed information regarding some connected topics. In order to preserve the natural flow of our presentation, these topics have made the object of a series of Appendices:

- Appendix A - **Fisher information for indirectly observed SMGRF textures** gives a detailed description on how to compute the Fisher information regarding the various parameters of the observation system and of the non-Gaussian texture model.
- Appendix B - **Optimal Bayesian Estimation** analyzes in parallel the optimality property of the Minimum Mean Squared Error estimator and of the evidence-based classifier, analysis relying on the Bayesian risk minimization.
- Appendix C - **Potts Model** explicits the form of the Potts prior for modeling the pixel interaction in image processing.
- Appendix D - **Truncation Matrices** explains in detail the structure, the connection

with the label field and the role of these matrices used in [Chapter 6](#).



# Inverse Problems, Bayesian Framework and Stochastic Sampling

## Contents

<b>2.1 Inverse Problems in Image Processing</b>	<b>7</b>
2.1.1 Indirect Data	8
2.1.2 Full Inversion Problem	10
<b>2.2 Bayesian Approach – Prior, Joint and Posterior Laws</b>	<b>11</b>
2.2.1 Uninformative Priors. Conjugacy	12
2.2.2 Estimation	12
<b>2.3 Addressed Problems</b>	<b>13</b>
<b>2.4 Efficient Metropolis-Hastings Samplers</b>	<b>13</b>
2.4.1 Independent	15
2.4.2 Standard Random-Walk	16
2.4.3 Langevin adapted Random-Walk	16
2.4.4 Hessian adapted Random-Walk	17
2.4.5 Fisher adapted Random-Walk	18
<b>2.5 Conclusion</b>	<b>20</b>

## 2.1 Inverse Problems in Image Processing

Inverse problem is the denomination given to any attempt to infer on the inputs of a physical system, starting from the data observed at its output.



Some of the most commonly encountered inverse problems in signal and image processing are deconvolution, denoising, parameter estimation, segmentation, super-resolution, optical flow estimation, source detection, particle image velocity, model choice.

In general, the unknowns are obtained as the solution of a system with  $N$  equations, where  $N$  is the number of observations,  $O$ . Most often than not, they are ill-posed problems, since the number of data is smaller than the number of unknowns and thus it is not

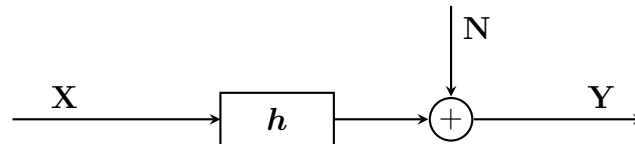
straightforward to determine a unique solution. In order to overcome this impasse, supplementary constraints must be imposed on the solutions. This is done by introducing information concerning the structure of the expected solution. This information is called regularization information, it is acquired independently on the observation process and it represents the knowledge we possess concerning the physical quantity. This information can introduce for instance a smoothness or a sparsity constraint, a restriction of the range of possible values or favor certain solutions.

The Bayesian framework represents a natural setting for such problems. Its principle consists in probabilizing the unknowns and defining a likelihood law and *a priori* laws. The likelihood is formulated based on the direct physical model of the system and represents the probability density of the data realizations, conditionally on the unknowns. The *a priori* laws contain the regularization information under the form of probability distributions for the unknowns. The solution to the inverse problem can be found by maximizing a criterion based on two terms: a data term, corresponding to the likelihood law in the Bayesian setting, and the regularization term.

Among the numerous and varied aforementioned problems, this work focuses on **parameters estimation, model choice and segmentation**, all this in the context of **indirect observations**. The indirect observations mean that the data are a blurred and noisy version of the original image on which we should base the further inference to determine the quantities of interest. Consequently, throughout this work, deconvolution is a recurrent topic, bringing an additional layer of complexity to the problems. These problems are exclusively addressed in the advantageous Bayesian framework, where the aspect of indirect data can naturally be merged with the actual problem of estimating the underlying parameters, selecting the most probable model or segmenting the original image.

### 2.1.1 Indirect Data

Deconvolution is a topic generating vivid interest in the signal processing community. It is encountered in most applications involving indirect observations, since there are no ideal observation systems. Let us represent the convolution based observation system as follows:



This convolutive system is described by the equation:

$$\mathbf{Y} = \mathbf{h} * \mathbf{X} + \mathbf{N} \quad (2.1)$$

where  $\mathbf{X}$ ,  $\mathbf{N}$  and  $\mathbf{Y}$  are the unobserved original image, the noise and the observations, respectively, and  $\mathbf{h}$  is the convolution filter. The original images are of size  $N \times N$  and  $P = N^2$  is the total number of pixels. In the following, the mathematical developments

are based on the alternative expression of (2.1):

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (2.2)$$

where the  $P \times 1$  size vectors  $\mathbf{x}$ ,  $\mathbf{n}$  and  $\mathbf{y}$  are obtained by lexicographically ordering the pixels of the  $\mathbf{X}$ ,  $\mathbf{N}$  and  $\mathbf{Y}$  matrices. Furthermore,  $\mathbf{H}$  represents the  $P \times P$  convolution matrix representing the point spread function (PSF) of the blurring filter.

In the context of such an observation system, various problems may be considered:

- for a known PSF (known  $\mathbf{H}$ ), if the eigenvalues of  $\mathbf{H}$  are not tending to zero, *i.e.*, there are no instabilities, the problem is well-posed and can be solved by applying the inverse filter. When dealing with **noise corrupted data**, even for known PSF, the problem becomes ill-posed. Nevertheless, it can be solved through Wiener filtering.
- for an **unknown PSF** (unknown  $\mathbf{H}$ ), the instrument characteristic is estimated along with the original image. This ill-posed problem can be solved by adding regularization information in order to constrain the solution. One possibility is to make no assumptions on the form of  $\mathbf{H}$ , except for some regularity constraints, meaning that the PSF estimation implies the estimation of all its elements – blind deconvolution. The alternative option is to consider that the PSF has a **parametric form** ( $\mathbf{H}_\eta$ ), in which case its estimation comes down to estimating the parameters ( $\eta$ ) driving this form – **myopic / semi-blind deconvolution**.

The highlighted aspects describe the observation system considered in this work: data corrupted by noise and an unknown, parametric PSF, driven by the parameter set  $\eta$ .

In this context, we have focused on an additive, Gaussian model for the noise. This model is driven by the covariance  $\mathbf{R}_n(\gamma_n)$ , which depends on the parameters  $\gamma_n$ . Although the methods presented in the following are adapted for any type of additive Gaussian noise, for algorithmic simplicity, we are only considering the case of white noise. This is the assumption made in the vast majority of the deconvolution applications [MMK06, BMK08, BMK09, BWMK10, BMK11] and consists in setting  $\mathbf{R}_n = \gamma_n \mathbf{I}$ , where  $\gamma_n$  is the inverse variance / precision parameter for the noise. In this case, based on the aforementioned assumptions about the noise, the likelihood law can be expressed as:

$$f(\mathbf{y}|\gamma_n, \eta, \mathbf{x}) = (2\pi)^{-P/2} \gamma_n^{P/2} \exp \left[ -\frac{1}{2} \gamma_n \|\mathbf{y} - \mathbf{H}_\eta \mathbf{x}\|^2 \right] \quad (2.3)$$

The operation of convolution has the special property of being written as a product in the Fourier domain. Moreover, in the case where the  $\mathbf{H}_\eta$  has a circulant form, it is diagonalizable by Discrete Fourier Transform (DFT). This assumption has minimal impact when the size of the image is large with respect to the support of the convolution filter, which is true in the case of most observation systems.

Let us denote by  $\hat{\mathbf{y}}$ ,  $\hat{\mathbf{n}}$ ,  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{h}}$  the DFTs of the observations, noise, original textured image and PSF, respectively. Based on the white noise and circulant PSF assumptions, the law for the noise in the Fourier domain is separable, making way for significant mathematical simplifications and, consequently, performance enhancements. From the noise law,

the likelihood can be straightforwardly deduced and writes:

$$f(\mathring{\mathbf{y}}|\gamma_n, \boldsymbol{\eta}, \mathring{\mathbf{x}}) = (2\pi)^{-P} \gamma_n^P \exp \left[ -\gamma_n \sum_{p=1}^P |\mathring{y}_p - \mathring{h}_p(\boldsymbol{\eta}) \mathring{x}_p|^2 \right] \quad (2.4)$$

where  $\mathring{h}_p(\boldsymbol{\eta})$  shows explicitly the parametric form of the Transfer Function (TF), the frequency domain counterpart of the PSF.

In order to exploit this law separability, the mathematical developments are done in the frequency domain. Consequently, we will be dealing with complex quantities:  $\mathring{\mathbf{y}}, \mathring{\mathbf{n}}, \mathring{\mathbf{x}}, \mathring{\mathbf{h}} \in \mathbb{C}^P$ . Moreover, for model simplicity and computational efficiency, we have also considered that the spatial domain variables are complex:  $\mathbf{y}, \mathbf{n}, \mathbf{x}, \mathbf{h} \in \mathbb{C}^P$ . In this manner, the supplementary conditions of spectrum symmetry are avoided.

**Remark:** The expression in Equation (2.4) takes into account the fact that we are dealing with complex quantities and has the form of a complex Gaussian.

This likelihood law describes our direct problem and is the basis of the inference to determine the unknown quantities. Nevertheless, it is straightforward that this law is not sufficient for the estimation, since the number of unknowns is higher than the number of observations. For this reason, regularization is necessary in order to determine a unique solution. In the following, we will give more detailed considerations concerning the regularization and the estimation process.

## 2.1.2 Full Inversion Problem

Let us consider the original image  $\mathbf{x}$  is composed of one or several regions  $R$ . Each of these regions consists in a single stationary texture patch, belonging to one of the  $K$  classes of textures. No assumption is made on the relation between  $R$  and  $K$ , since none of them is probabilized.  $K$  is known and  $R$  is not represented in the mathematical developments.

In order to fully describe the original image  $\mathbf{x}$ , a set of hidden variables (the hidden label field)  $\mathbf{z}$  can be introduced. The variable  $z_p = k$  indicates for each position  $p$  that the pixel  $x_p$  is extracted from the full texture  $\mathbf{x}_k$ . Consequently, the prior for the image can be written as  $f(\mathbf{x}|\mathbf{z}, \mathbf{x}_1, \dots, \mathbf{x}_K)$ .

Let  $k = 1 \dots K$  be the model index. Then, each texture  $\mathbf{x}_k$  can be fully described by its model, indexed by the discrete variables  $\mathcal{M}_k$ , and the corresponding model parameters  $\boldsymbol{\chi}_k$ . Further details regarding the texture models employed in this work are given in Chapter 3, for the moment the prior law for each textured image being written generically as  $f(\mathbf{x}_k|\mathcal{M}_k, \boldsymbol{\chi}_k)$ . In fact,  $\mathbf{x}$  is obtained deterministically from  $\mathbf{x}_1 \dots \mathbf{x}_K$  based on the variables  $\mathbf{z}$ . For this reason,  $\mathbf{x}$  is not represented in the variable hierarchy represented in Figure 2.1.

This global hierarchical model for our complete problem is based on the aforementioned interdependencies and the observation model presented in the previous section.

By analyzing this hierarchical dependency, the elevated complexity of this problem is obvious. Starting from the observations  $\mathbf{y}$ , the goal is to estimate:

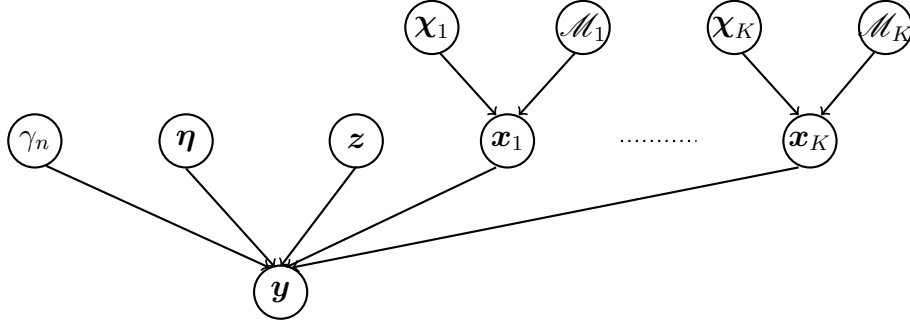


Figure 2.1: Hierarchical model for the complete problem.

- the unknowns of the observation system  $\gamma_n$  and  $\eta$ ,
- the original image  $x$ ,
- the hidden labels  $z$ ,
- the model  $\mathcal{M}_k$  and parameters  $\chi_k$  of the textures  $x_k$ .

This problem can only be solved by imposing a series of constraints, this being a typical procedure in the field of inverse problems. We have chosen to use a Bayesian approach, where the regularization is achieved through the priors on the parameters.

## 2.2 Bayesian Approach – Prior, Joint and Posterior Laws

The Bayesian approach [GCSR04, Stu10] consists in writing the *a posteriori* law for all the unknowns, given the observations, and using this law in order to obtain estimates for the unknowns. The posterior law can be determined from the joint law:

$$f(\gamma_n, \eta, z, x_{1...K}, \mathcal{M}_{1...K}, \chi_{1...K} | y) = \frac{f(y, \gamma_n, \eta, z, x_{1...K}, \mathcal{M}_{1...K}, \chi_{1...K})}{f(y)} \quad (2.5)$$

$$\propto f(y, \gamma_n, \eta, z, x_{1...K}, \mathcal{M}_{1...K}, \chi_{1...K})$$

This law, based on the variable dependency presented in Figure 2.1, writes:

$$f(y, \gamma_n, \eta, z, x_{1...K}, \mathcal{M}_{1...K}, \chi_{1...K}) = f(y | \gamma_n, \eta, z, x_{1...K}) \cdot \pi(\gamma_n) \cdot \pi(\eta) \quad (2.6)$$

$$\cdot \pi(z) \cdot \prod_k f(x_k | \mathcal{M}_k, \chi_k)$$

$$\cdot \prod_k \pi(\mathcal{M}_k) \cdot \prod_k \pi(\chi_k)$$

In order to fully specify this law, *a priori* laws for the unknowns must be chosen, these priors having the role of regularizing the problem. These laws must be completely unrelated to the observations and must reflect general knowledge about the unknowns.

Regarding the hidden labels, the only available prior information is that, irrespective of the observations, the pixels exhibit an aggregation tendency. This means that the labels have a tendency of forming regions with the same value. This can be formalized by using a Potts model for the labels (see Appendix C), with a temperature parameter  $\beta$  that drives the

strength of this aggregation behavior. The value of the  $\beta$  parameter determines the mean size of the regions of pixels with the same label.

### 2.2.1 Uninformative Priors. Conjugacy

The previous considerations concerning the labels have illustrated the influence of the priors on the entire estimation process. Consequently, in cases we do not possess prior information regarding the parameters, in order to avoid biasing the estimation, it is preferable to use uninformative priors. This is the case of  $\gamma_n$ ,  $\boldsymbol{\eta}$ ,  $\beta$ ,  $\mathcal{M}_{1\dots K}$  and  $\boldsymbol{\chi}_{1\dots K}$ .

Nevertheless, keeping in mind that the posterior law is used to determine the estimates, it is important from a computational point of view for this law to have standard forms. For this reason, when possible, the priors should have a conjugate form with respect to the likelihood or the other models depending on that parameter. For instance,  $\gamma_n$  is the precision parameter in the likelihood law. The conjugate form with respect to this dependency is a Gamma law, *i.e.*,:

$$\pi(\gamma_n|\alpha_n, \beta_n) = \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \gamma_n^{\alpha_n-1} \exp -\beta_n \gamma_n = \mathcal{G}(\alpha_n, \beta_n) \quad (2.7)$$

where  $\alpha_n, \beta_n$  are considered known. This distribution becomes uninformative in the limit case when  $\alpha_n \rightarrow 0$  and  $\beta_n \rightarrow 0$ . In this case, the Gamma distribution becomes an uninformative Jeffreys law  $\pi(\gamma_n) = 1/\gamma_n$ .

**Remark:** The problem in this case is that the Jeffreys law is an improper prior (it is not integrable), which may lead to an improper posterior and may give rise to problems in the estimation process. Nevertheless, in some situations the posterior law is proper despite the presence of improper priors [KW96], this being an open topic of research.

The priors for the rest of the unknowns will be defined in the following chapters, once the texture models  $f(\mathbf{x}_k|\mathcal{M}_k, \boldsymbol{\chi}_k)$  and the dependency  $\mathbf{h}(\boldsymbol{\eta})$  are given explicitly, however, they will all have uninformative forms.

**Remark:** The uninformative priors might give the wrong impression that our ill-posed problem is not regularized. In fact, this is done through highly structured models for the PSF, image and labels and not through the priors on the parameters.

Moreover, without anticipating too much, we can assert that the posterior law has a complicated form and its dependency with respect to some of the parameters does not have a standard form.

### 2.2.2 Estimation

The information contained by the aforementioned posterior law concerning the parameter values can be exploited using various estimators: Maximum *A Posteriori* (MAP), marginalized MAP (mMAP), Median *A Posteriori* (MeAP) or Posterior Mean (PM).

Among these estimators, the PM is the optimal one from the Mean Squared Error (MSE) point of view. This is in fact the reason for which we have chosen it among the various options. Similarly, the evidence based classifier is optimal from the mean classification risk viewpoint. The mathematical proof of these properties is given in Appendix B, where we have analyzed in parallel the two estimators.

## 2.3 Addressed Problems

The previous sections have presented a very complex inverse problem of myopic deconvolution and segmentation of an indirectly observed image consisting in several textured regions. Our goal is to devise a method that can achieve this and at the same time estimate the noise and signals levels, the model and the parameters of each of the textures present in the image. This is a very difficult task and thus we have divided it into three sub-problems:

- i. myopic deconvolution and parameter estimation of a textured image;
- ii. model selection and parameter estimation for a blurred and noisy textured image;
- iii. deconvolution segmentation and parameter estimation for a blurred and noisy image composed of different textures.

All these methods, which will be detailed in the following chapters, rely on formulating the posterior law and extracting the information encoded by this law. In every case the form of the posterior will be too complicated, thus numerical methods will be employed. Among the available options, we have chosen to use a Monte Carlo Markov Chain (MCMC) method, more specifically, Gibbs sampling. This iterative method sequentially samples every variable, conditionally on the values of the rest of the variables. In this context, in cases where the conditional *a posteriori* law for a certain variable has a non-standard form, more advanced samplers must be embedded within the Gibbs algorithm.

The following section is devoted to the analysis of several numerical methods and the presentation of our algorithmic contribution.

## 2.4 Efficient Metropolis-Hastings Samplers

One of the most commonly employed samplers for complicated laws is the Metropolis-Hastings (MH) algorithm. Then, the solution is to integrate an MH step for  $\theta$  in the Gibbs sampler. This Metropolis within Gibbs strategy is convergent, as proven by [Tie94], thus provides samples from the posterior distribution, in our case  $\pi(\Psi|\mathbf{y})$ . Since the performances of the sampler directly affect the estimation process, the following section is devoted to the study of efficient versions of the MH algorithm.

The recent literature on efficient sampling is abundant and includes methods that rely either on optimal tuning of the standard MH samplers [HST01, Ros11], or on formulating the proposal law based on the target in order to achieve an efficient exploration.

An MH algorithm, as described by [MRR<sup>+</sup>], [Has70], relies on a transition kernel



consisting of two ingredients: a transition law,  $q(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p)$ , and an acceptance probability:

$$\alpha(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p) = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_p)}{\pi(\boldsymbol{\theta}_c)} \cdot \frac{q(\boldsymbol{\theta}_p, \boldsymbol{\theta}_c)}{q(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p)} \right\} \quad (2.8)$$

where  $\boldsymbol{\theta}_c$  is the current and  $\boldsymbol{\theta}_p$  is the proposed value for the parameters and let  $\Xi = \frac{q(\boldsymbol{\theta}_p, \boldsymbol{\theta}_c)}{q(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p)}$  be the ratio of the jumps. The functioning of an MH algorithm is the following:

1. Initialize the iteration counter  $j = 1$  and set  $\boldsymbol{\theta}_c^{(0)}$ .
2. Propose a new value  $\boldsymbol{\theta}_p$  for the parameter, generated from the density  $q(\boldsymbol{\theta}_c^{(j-1)}, \cdot)$ .
3. Evaluate the acceptance probability  $\alpha(\boldsymbol{\theta}_c^{(j-1)}, \boldsymbol{\theta}_p)$  given by Equation (2.8) and according to this value accept the proposal and update the parameter  $\boldsymbol{\theta}_c^{(j)} = \boldsymbol{\theta}_p$ , or reject it and keep the old value for the parameter  $\boldsymbol{\theta}_c^{(j)} = \boldsymbol{\theta}_c^{(j-1)}$ .
4. Update the counter and return to step 2 until convergence.

There are numerous options for formulating the proposal and the convergence speed and mixing properties are directly influenced by the adequacy between the proposal law and the target. The simplest version of this algorithm is the Independent MH (IMH), whose proposal does not depend on the current value  $\boldsymbol{\theta}_c$ . Nevertheless, this algorithm's simplicity is reflected in its inability to use any of the previously acquired information to render the sampling more efficient. For this reason, its use is limited to cases where the target has very complicated forms with mass distributed in a large area of the parameter space.

Except for the "independent" version of this algorithm, the other versions can be considered as being Random Walks (RWMH), since their proposals contain the current value of the chain  $\boldsymbol{\theta}_c$  and a stochastic term. The simplest version is the isotropic RWMH, which proposes an isotropic displacement around the current value of the chain.

Moreover, choosing a transition kernel  $q$  that embeds information about the shape of the target can significantly enhance the algorithm performances, especially when the target is very peaked, as in our problem. Advanced methods include a directional component for the proposal, meaning that the proposal is built by making an isotropic move around the current value plus the directional component. In the class of directional MH methods, a first idea is to build the proposal based on first order derivatives of the target, this being the case of the Metropolis Adjusted Langevin Algorithm (MALA) [RS03, KKS<sup>+</sup>10, GC11, MR12] and of the Hamiltonian methods [GC11, ZS11, Nea11, BPSS11]. Another class of methods that improve the sampling performances is the quadratic (Newton-like) approach, based on second order derivatives (Hessian matrices for multidimensional parameter spaces) and exploiting the information regarding the target curvature. Such methods have first been employed in optimization theory, but they have recently been successfully adapted to sampling [QM02, BTG12, MWBG12].

In this work, we have focused our attention on the second order derivative-based proposals, which we have improved by building a new transition kernel based on the Fisher information matrix. Our contribution is modifying the Hessian algorithm presented in [QM02] to eliminate the concerns regarding the positiveness of the Hessian matrix. This is achieved by its replacement with the Fisher information matrix. The results obtained for the proposed problem are encouraging, confirming the stable behavior and speed perfor-



mances of the second order derivative-based samplers.

All of the analyzed samplers are convergent and accurately explore the target, the differences between them being given by the different formulation of the transition kernel and, implicitly, by the expression of the acceptance probability. These differences translate into:

- different exploration of the target, this influencing how fast the sampler reaches the high probability regions,
- differences between the time needed by each sampler to produce one sample.

In the following, we will present the specific aspects of these algorithms, adapted to our inverse problems. The target law for these algorithms is the *a posteriori* law for some of the texture parameters  $\theta$ . In our case, this law has a very peaked exponential form and its values may overpass Matlab's numerical precision. For this reason, the evaluation of the acceptance probability can lead to numerical problems and indeterminations.

By applying the logarithm to Equation (2.8) we avoid handling exponential quantities. Consequently, in order to evaluate this probability, we only have to compute a difference between the Log-Posterior (LP) taken in the proposed value and in the current value, where  $LP = \log \pi(\theta|\mathbf{y})$ . Moreover, instead of computing the second ratio in Equation (2.8), it suffices to compute the value of  $\Xi$ :

$$\log \alpha = \min \{0, LP(\theta_p) - LP(\theta_c) + \log \Xi\} \quad (2.9)$$

Furthermore, let us consider that the prior law for  $\theta$  is uniform  $\pi(\theta) = \mathcal{U}_{[\theta_m, \theta_M]}(\theta)$ . We can then write:

$$LP(\theta) = LL(\theta) + \log \pi(\theta) - \log f(\mathbf{y}) \quad (2.10)$$

where LL denotes the Log-Likelihood, and  $\log \pi(\theta) = -\log(\theta_M - \theta_m)$  and  $\log f(\mathbf{y})$  are constant with respect to  $\theta$ . Then, (2.9) can be rewritten as:

$$\log \alpha = \min \{0, LL(\theta_p) - LL(\theta_c) + \log \Xi\} \quad (2.11)$$

In the following, we will present the various versions of the MH algorithm and their specificities in terms of computational complexity and efficient parameter space exploration.

### 2.4.1 Independent

The proposal for this algorithm writes:

$$\theta_p = \mathcal{U}_{[\theta_m, \theta_M]}(\theta) \quad (2.12)$$

where the values of  $\theta_m, \theta_M$  are the limits of the parameter space. In this case, the transition kernel is independent of the current value  $\theta_c$  and writes:

$$q(\theta_c, \theta_p) = \mathcal{U}_{[\theta_m, \theta_M]}(\theta_p) \quad (2.13)$$

The acceptance probability becomes:

$$\begin{aligned}\log \alpha &= \min \left\{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) + \log [\mathcal{U}_{[\boldsymbol{\theta}_m, \boldsymbol{\theta}_M]}(\boldsymbol{\theta}_c)] - \log [\mathcal{U}_{[\boldsymbol{\theta}_m, \boldsymbol{\theta}_M]}(\boldsymbol{\theta}_p)] \right\} \\ &= \min \{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) \}\end{aligned}\quad (2.14)$$

since the transition kernel is uniform and has the same value for any  $\boldsymbol{\theta}$ .

As shown by the previous equations, this algorithm consists in very simple and inexpensive iterations. Nevertheless, in most cases this proposal is not adapted to the form of the target, leading to high rejection rates and, consequently, to very slow convergence.

### 2.4.2 Standard Random-Walk

Taking into account the information that the current value  $\boldsymbol{\theta}_c$  has been considered adequate enough to be accepted can improve the performances of the MH sampler. This information is embedded in the proposal, which becomes:

$$\boldsymbol{\theta}_p = \boldsymbol{\theta}_c + \varepsilon \mathcal{N}(0, \mathbf{I}) \quad (2.15)$$

In this case, the value of  $\varepsilon$  sets the size of the jumps. This is very important since a small value of the step results in a slow exploration of the parameter space and strongly correlated samples, while a large value may lead to high rejection rates, as in the case of the IMH. The optimal value for  $\varepsilon$ , in the sense that it provides the best compromise between the amplitude of the transitions and the acceptance rate is the one that yields an acceptance rate of approximately 24% [GRG96].

In this case, the transition kernel writes:

$$q(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p) = \exp \left[ -\frac{1}{2\varepsilon^2} \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_c\|^2 \right] \quad (2.16)$$

The acceptance probability becomes:

$$\begin{aligned}\log \alpha &= \min \left\{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) - \frac{1}{2\varepsilon^2} \|\boldsymbol{\theta}_c - \boldsymbol{\theta}_p\|^2 + \frac{1}{2\varepsilon^2} \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_c\|^2 \right\} \\ &= \min \{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) \}\end{aligned}\quad (2.17)$$

due to the symmetry of the transition kernel.

The isotropic RWMH algorithm also has a reduced cost per iteration. Nevertheless, depending on the initialization and the value of the tuning parameter, it may be slow to converge. Moreover, it does not exploit the available information concerning the form of the target in order to increase sampling efficiency.

### 2.4.3 Langevin adapted Random-Walk

Langevin algorithms are derived from diffusion approximations and rely on the principle of using the information concerning the target density, in the format  $\nabla \log \pi$ , in order

to build a proposal distribution well-adapted to the problem in question [GRG96]. The Langevin-based MH algorithm proposes an RW-like transition of the form [BGHM95]:

$$\boldsymbol{\theta}_p = \boldsymbol{\theta}_c - \frac{\varepsilon^2}{2} \mathbf{g}(\boldsymbol{\theta}_c) + \varepsilon \mathcal{N}(0, \mathbf{I}) \quad (2.18)$$

where  $\mathbf{g}(\boldsymbol{\theta}_c) = \frac{\partial \text{LL}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_c}$ . The acceptance probability can be obtained from Equation (2.11), by using the transition kernel:

$$q(\boldsymbol{\theta}_p, \boldsymbol{\theta}_c) = \exp \left[ -\frac{1}{2\varepsilon^2} \left\| \boldsymbol{\theta}_c - \boldsymbol{\theta}_p - \frac{\varepsilon^2}{2} \mathbf{g}(\boldsymbol{\theta}_p) \right\|^2 \right] \quad (2.19)$$

Then, this acceptance probability writes:

$$\log \alpha = \min \left\{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) - \frac{1}{2\varepsilon^2} \left[ \left\| \boldsymbol{\theta}_c - \boldsymbol{\theta}_p - \frac{\varepsilon^2}{2} \mathbf{g}_p \right\|^2 - \left\| \boldsymbol{\theta}_p - \boldsymbol{\theta}_c - \frac{\varepsilon^2}{2} \mathbf{g}_c \right\|^2 \right] \right\}$$

and the log ratio of the jumps becomes:

$$\log \Xi = -\frac{1}{2} \left[ (\boldsymbol{\theta}_p - \boldsymbol{\theta}_c)^t (\mathbf{g}_p - \mathbf{g}_c) + \frac{\varepsilon^2}{2} (\mathbf{g}_p^t \mathbf{g}_p - \mathbf{g}_c^t \mathbf{g}_c) \right] \quad (2.20)$$

As compared to the non-directional MH methods, the complexity of the Langevin MH is increased due to the form of the acceptance probability and to the necessity of evaluating the gradient for every new proposal. However, the increased computation time per iteration is compensated by the smaller number of iterations needed to reach convergence.

In regions far from the maximum of probability, the gradient is large (the directional component is dominant), *i.e.*, the algorithm approaches the high probability regions with high amplitude jumps. Near the maximum of probability, the gradient is small, thus the stochastic component is dominant and it permits for the region to be explored.

#### 2.4.4 Hessian adapted Random-Walk

This section is devoted to a sampling method seldom explored and whose presence in the literature is scarce. The directional component of the proposal is in this case formulated using Newton's direction, which, for a quadratic law, indicates the maximum. In [QM02] a version of this sampler has been tested and compared to methods such as Gibbs and optimal marginal data augmentation (DA) samplers on a probit regression problem, proving that the performances of this sampler are superior. The transition for the Hessian sampler is of the form:

$$\boldsymbol{\theta}_p = \boldsymbol{\theta}_c + \varepsilon \boldsymbol{\Sigma}(\boldsymbol{\theta}_c) \mathbf{g}(\boldsymbol{\theta}_c) + \mathcal{N}(0, \boldsymbol{\Sigma}(\boldsymbol{\theta}_c)) \quad (2.21)$$

where  $\boldsymbol{\Sigma}(\boldsymbol{\theta}_c) = -\mathbf{H}(\boldsymbol{\theta}_c)^{-1}$  and  $\mathbf{H}(\boldsymbol{\theta}_c)$  is the Hessian matrix of the target law, computed in the value  $\boldsymbol{\theta}_c$ . The acceptance probability is obtained from (2.11), for:

$$q(\boldsymbol{\theta}_p, \boldsymbol{\theta}_c) = \mathcal{N}(\boldsymbol{\theta}_c - \boldsymbol{\theta}_p + \varepsilon \boldsymbol{\Sigma}(\boldsymbol{\theta}_p) \mathbf{g}(\boldsymbol{\theta}_p), \boldsymbol{\Sigma}(\boldsymbol{\theta}_p)) \quad (2.22)$$

In this case, the acceptance probability has the form:

$$\log \alpha = \min \left\{ 0, \text{LL}(\boldsymbol{\theta}_p) - \text{LL}(\boldsymbol{\theta}_c) - \frac{1}{2} [\log (\det (\boldsymbol{\Sigma}(\boldsymbol{\theta}_p))) - \log (\det (\boldsymbol{\Sigma}(\boldsymbol{\theta}_c)))] \right. \\ \left. - \frac{1}{2} \|\boldsymbol{\theta}_c - \boldsymbol{\theta}_p + \varepsilon \boldsymbol{\Sigma}(\boldsymbol{\theta}_p) \mathbf{g}(\boldsymbol{\theta}_p)\|_{\boldsymbol{\Sigma}(\boldsymbol{\theta}_p)}^2 + \frac{1}{2} \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_c + \varepsilon \boldsymbol{\Sigma}(\boldsymbol{\theta}_c) \mathbf{g}(\boldsymbol{\theta}_c)\|_{\boldsymbol{\Sigma}(\boldsymbol{\theta}_c)}^2 \right\}$$

and the log ratio of the jumps in this case writes:

$$\log \Xi = -\frac{1}{2} [\log (\det (\boldsymbol{\Sigma}(\boldsymbol{\theta}_p))) - \log (\det (\boldsymbol{\Sigma}(\boldsymbol{\theta}_c)))] - \frac{1}{2} \left\{ \|\boldsymbol{\theta}_c - \boldsymbol{\theta}_p\|_{\boldsymbol{\Sigma}_p - \boldsymbol{\Sigma}_c}^2 \right. \\ \left. - 2\varepsilon (\boldsymbol{\theta}_c - \boldsymbol{\theta}_p)^t (\mathbf{g}_p + \mathbf{g}_c) + \varepsilon^2 [\mathbf{g}_c^t \boldsymbol{\Sigma}(\boldsymbol{\theta}_c) \mathbf{g}_c - \mathbf{g}_p^t \boldsymbol{\Sigma}(\boldsymbol{\theta}_p) \mathbf{g}_p] \right\} \quad (2.23)$$

The advantage of the method is that, for quadratic and quasi-quadratic distributions, the regions of high probability are approached in a very small number of iterations (ideally, a single one) and then explored with the contribution of the stochastic component, a quadratic law of variance  $\boldsymbol{\Sigma}(\boldsymbol{\theta}_c)$ , which is an accurate approximation of the target. However, it is clear that this method is also rather complex, as each iteration translates in the computation of the gradient and the Hessian matrix and the evaluation of the acceptance probability.

In the context of quadratic directional methods, a major concern is the need to perform the inversion of the Hessian, as this may be problematic if the Hessian is not positive definite.

### 2.4.5 Fisher adapted Random-Walk

In order to avoid the aforementioned matrix inversion problems, in the present work we have explored the idea of replacing the Hessian by the Fisher information matrix (see Appendix A), which is by definition positive definite:

$$\mathcal{J}_{pq}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}|\boldsymbol{\theta}} \left[ -\frac{\partial^2}{\partial \theta_p \partial \theta_q} \text{LL}(\boldsymbol{\theta}) \middle| \boldsymbol{\theta} \right] \quad (2.24)$$

The Fisher matrix quantifies the mean amount of information that the observations contain regarding the parameter  $\boldsymbol{\theta}$ . We have been able to apply such an approximation, as in the case of our problem, we have a great amount of independent observations, thus, a scenario close to the asymptotic case. This efficient sampler has been presented in [VGB11] under the name of Fisher adapted MH (FMH).

A very interesting sampler, developed independently from our own is the manifold MALA (mMALA) [GC11], dedicated to highly-dimensional and highly-correlated targets. Its development is based on the symmetric Kullback Leibler divergence,  $\mathcal{D}_S(p||q)$ , a first order Taylor approximation,  $f(\mathbf{y}|\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx f(\mathbf{y}|\boldsymbol{\theta}) + \delta\boldsymbol{\theta}^t \nabla_{\boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta})$  and the approximation

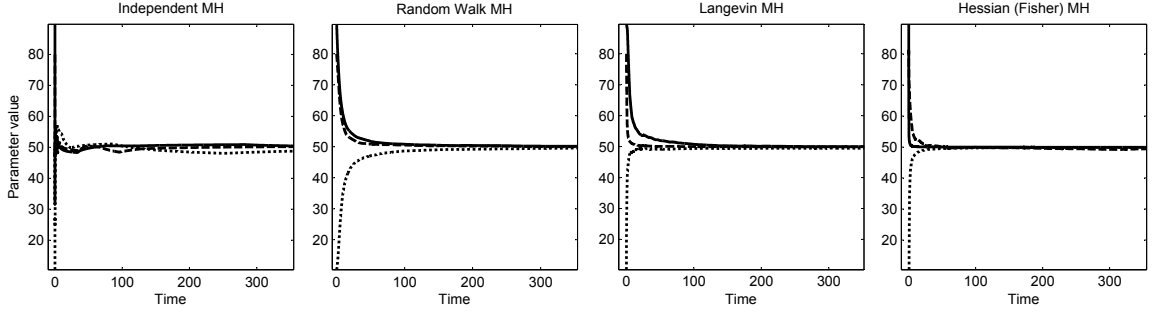


Figure 2.2: The stabilization of the PM for several chains in the case of each of the four studied methods. Each chain corresponds to a different initialization. As expected, all the algorithms converge to the same value, but the FMH reaches equilibrium the fastest. Although superior in terms of computation speed per iteration, the IMH and RWMH require a longer interval to converge.

$\log(1 + \varepsilon) \approx \varepsilon$ :

$$\begin{aligned} \mathcal{D}_S[f(\mathbf{y}|\boldsymbol{\theta} + \delta\boldsymbol{\theta})||f(\mathbf{y}|\boldsymbol{\theta})] &= \\ &= \delta\boldsymbol{\theta}^t \cdot \underbrace{\mathbb{E}_{\mathbf{y}|\boldsymbol{\theta}} [\nabla_{\boldsymbol{\theta}} \log f(\mathbf{y}|\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} \log f(\mathbf{y}|\boldsymbol{\theta})^t]}_{\mathcal{J}(\boldsymbol{\theta})} \cdot \delta\boldsymbol{\theta} \end{aligned} \quad (2.25)$$

The mMALA is obtained by defining the Langevin diffusion with invariant measure  $\pi(\boldsymbol{\theta}|\mathbf{y})$ . This algorithm formulates the proposal:

$$\boldsymbol{\theta}_{\text{pr}} = \boldsymbol{\theta}_c + \frac{\varepsilon^2}{2} \mathcal{J}^{-1}(\boldsymbol{\theta}_c) \cdot \nabla_{\boldsymbol{\theta}} \text{LP}(\boldsymbol{\theta}_c) + \varepsilon \sqrt{\mathcal{J}^{-1}(\boldsymbol{\theta}_c)} \cdot \mathbf{z}_c \quad (2.26)$$

with  $\text{LP}(\boldsymbol{\theta}) = \log \pi(\boldsymbol{\theta}|\mathbf{y})$  the log-posterior and  $\mathbf{z}_c \sim \mathcal{N}(0, \mathbf{I})$  an isotopic displacement.

Our FMH algorithm, that uses the same proposal law as the one in Equation (2.26), is in fact based on the idea of quasi-Newton proposals [QM02] for a fast exploration of the parameter space and superior mixing properties and replacing the Hessian with the Fisher information matrix. The use of this proposal proved advantageous from multiple points of view. Firstly, this exploits the target curvature similarly to the Newton step from the optimization theory. Secondly, this made way for a series of algorithmic simplifications and performance enhancement:

- for our GRF modeled textures (described in Chapter 3), the second order derivatives vanish under the expectation. This allows the formulation of the efficient proposal only based on first order derivatives,
- the Fisher matrix is positive definite. Hence, if its eigenvalues are not zero, there are no instabilities when taking the inverse, such as those mentioned in [QM02] for the use of the Hessian,
- also due to the positive definite Fisher matrix, the Newton term always has the direction of gradient ascent, thus the algorithm only makes efficient steps.

Figures 2.2 and 2.3 illustrate the samples evolution in the case of the MH samplers

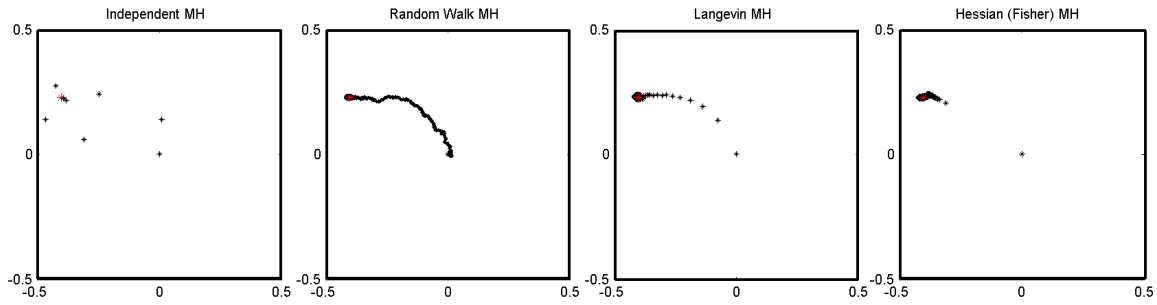


Figure 2.3: Samples evolution in a 2D parameter space for the four samplers. To the left, observe the sparsity of the accepted samples for IMH, for RWMH the evolution step is very small and undirected, while for Langevin MH the proposal is influenced by the gradient. For FMH, the strong probability regions are approached in a single iteration and then thoroughly sampled, as these are the regions most representative for the target.

previously presented. As expected, the quadratic methods require the least number of samples to converge. Moreover, due to the simplification achieved using the Fisher matrix, the overall convergence time is the most reduced for the FMH.

## 2.5 Conclusion

This chapter has presented the formulation of our full inverse problem, theoretical aspects regarding the Bayesian framework that is employed, its mathematical formulation and practical issues related to the implementation. We have specified our choice of estimator and the reason behind this choice. In this context, sampling is used to numerically compute the estimator and several versions of the MH sampler are presented.

Our contribution consists in developing the new efficient sampler FMH, based on the Fisher information matrix. This sampler will be employed in one of the inverse problems presented in the following chapters in order to enhance the speed of our sampling based method. As a perspective on this topic, this sampler can be integrated in numerous other sampling based applications in order to accelerate the sampling speed.

# Texture Modeling

## Contents

<b>3.1 Introduction and State of the art</b>	<b>21</b>
<b>3.2 Texture Modeling by Random Fields</b>	<b>25</b>
3.2.1 Gaussian Model	27
3.2.2 Non-Gaussian Model	32
<b>3.3 Conclusion and Perspectives</b>	<b>35</b>

Texture represents one of the main elements of the framework explored in this work. Inverse problems such as myopic deconvolution, parameter estimation, model choice, joint deconvolution + segmentation, of high interest in signal and image processing, are addressed in the special case of textures. Our work is set apart from the existing literature by the fact that we are not only dealing with texture, a rather challenging aspect in itself, but we are also considering indirect observations, thus introducing an additional complexity level.

Generally, ill-posed inverse problems are regularized through priors on the parameters. In this work, however, the highly structured image models (texture models) provide the needed constraints for the regularization.

## 3.1 Introduction and State of the art

Referring to **texture** in general, it is an omnipresent aspect in everyday life, the Oxford Dictionaries defining it as *"the feel, appearance, or consistency of a surface or a substance"*. Consequently, texture is one of the main characteristics of any material object. Moreover, the notion of texture has also been extended to more abstract quantities and has become a central aspect in signal processing. For instance, we can find a texture in an audio signal or in a data sequence. Nevertheless, most of the texture-related applications can be found in image processing, [SS01b] defining the **image texture** as the *"set of metrics calculated in image processing designed to quantify the perceived texture of an image"*. This is only one of the numerous definitions and descriptions that have been formulated in the literature, each influenced by the application, the author's background or the approach being employed.

Texture modeling represents an important aspect in image processing applications, the goal being to find comprehensive models that allow a good description, but are relatively

easy to handle and integrate in a more complex image processing formalism. Consequently, there is always a compromise to make between model versatility and complexity.

In the context of a growing interest towards texture in image processing applications, in the late 1990s several extensive works have tried to make an inventory of the texture types, approaches and applications. Amongst the most well-known, [TJ98] provides a valuable overview of texture definitions, incentives for taking interest in texture, fields of applications, types of modeling and image processing problems that can arise in the context of textured images. The authors classify the approaches to texture analysis into four main classes:

1. **statistical methods** – co-occurrence matrices, autocorrelation features,
2. **geometrical / structural methods** – Voronoi tessellation features,
3. **model based methods** – Random Field (RF) models, fractals,
4. **signal processing / transform-based methods** – spatial domain filters, Fourier domain filters, Gabor and wavelet methods.

This classification, also used in [MS98], has been enriched over time by the development of new modeling techniques. However, the main classes remain the same, as proven by its use in the more recent [BLM04].

A great amount of attention has been dedicated to **feature extraction**. For instance, [SS01a] provides an evaluation of five feature extraction methods: autocorrelation, edge frequency, primitive-length, Laws' method [Law80], and co-occurrence matrices. [SS08] offers a more comprehensive overview of statistical texture analysis. The authors include in their comparison transform-based methods such as Fourier, Gabor functions and wavelet transforms. Although in contradiction with the previous classification, which separates the statistical and the transform approaches, this point of view is coherent and generalizes the statistical approach beyond the spatial representation domain.

Texture modeling in a transformed space, as for instance the **wavelet** domain, has been explored for decades, [CJK93, AG03] pointing out the main advantage of the tree-structured wavelet transform that provides non-redundant representations, with wavelet coefficients that take at most the storage space of the original image. Moreover, as opposed to the Laws or Gabor filters, the filter coefficients do not require tuning and do not differ from one texture to another. The wavelet based approach has been used mostly in texture classification and retrieval applications, as is the case of [DV02b], where a signature is defined for each texture and the classification is based on the Kullback-Leibler distance, or of [DV02a], where a rotation invariant method is devised based on wavelet domain Hidden Markov Models (HMM). [FX03] uses a similar wavelet domain HMM principle and exploits the cross correlation across the sub-bands. A texture synthesis algorithm based on steerable pyramids with complex coefficients is given in [PS00].

In the same class of transform based methods, **frequency analysis** represents a very important approach. [Bro99] presents the idea behind Fourier transform based texture synthesis, *i.e.*, the fact that all stationary texture realizations should have the same amplitude spectrum. Since the phase spectrum localizes the frequency components in space, in the case of stationary textures, this can differ between the realizations. In the same line of



work, the recent [GGM11] shows two very interesting aspects: firstly, that "random phase textures and random shift textures generated from the same sample are indistinguishable" and, secondly, that "random phase textures are perceptually invariant under a multiplicative noise on the Fourier modulus". Moreover, texture features based on the Fourier transform can be very efficient in classification tasks, as shown in [ZFS01], where a feature based on histograms of local Fourier coefficient maps is tested for image retrieval and in [AKJ07], in a comparison with Gabor based features. In the dynamic textures field, [ACS05] reports a significant speed enhancement of the learning process and the need for a smaller learning set due to the use of a set of Fourier descriptors on the frames, instead of the raw sequence, for the linear dynamic system identification. [LLZ06] is also worth mentioning since it provides an interesting method for 2D texture synthesis starting from a 1D signal, by using the Fourier coefficients of the 1D signal and arranging them to form a so called pseudo Discrete Cosine Transform (DCT) and then applying and inverse DCT to obtain the texture.

Among the stochastic representations, the **Wold decomposition model** has also been used for texture modeling [SFP96]. It consists in decomposing the texture field into a sum of two orthogonal components: a deterministic and an indeterministic component. The deterministic component accounts for the structural properties, and is further decomposed into a harmonic and a finite number of evanescent components, while the structureless indeterministic component accounts for the randomness of the texture. [LP96] defines the main dimensions of human texture perception: "periodicity", "directionality" and "randomness", and provides a more robust method for texture modeling and classification. Moreover, [CNS00] shows that the indeterministic component can be predicted from the deterministic one using suitable nonlinear schemes. The more recent [STNR05] presents a parametric model for representing 3D textures, which describes both spatially and spectrally each component of the Wold decomposition.

The **statistical approaches** are not only adapted for texture analysis, by inferring on the underlying model parameters, but also for texture synthesis. This can be achieved by characterizing a set of visually similar textures through a probability distribution on a RF. It is a two step process, firstly, the feature extraction phase uses a filter set to capture the texture features characteristic to that class. The histograms of the filtered images are used to estimate the marginal distributions on the RF. Secondly, the feature fusion phase consists in determining the maximum entropy distribution that matches the previously determined marginals. This approach, based on the minimax entropy principle has been presented in [ZWM97] and then revisited in [ZWM98]. The maximum entropy based texture analysis has been employed in applications such as estimating the crystallite orientation distribution function [Boh05], texture and object recognition [LSP05].

In a more **structural oriented approach**, [MBLS01] tackles the image segmentation problem using a method based jointly on a texon based texture analysis and on contours. Another interesting application is the texture synthesis method in [WL00] inspired from the Markov RF (MRF) texture models and relying on a deterministic searching process.

The special properties of textures somewhat resemble those of fractals, which are defined by self-similarity and scale invariance. The theory of fractals has thus inspired the use of new descriptors for texture analysis, namely the fractal dimension and lacunarity

[CKC93]. The problem of texture segmentation based on the fractal dimension has been presented in [CS95], while the more recent [LH10] provides a robust and efficient method for computing the fractal dimension and its application in image classification. However, the fractal dimension characterizes self-similarity only in ideal cases, this being the reason for using multifractal analysis, this idea being explored in [XFZ06] for texture segmentation.

One of the most accessible methods, equally adapted to synthesis and analysis tasks, is **RF modeling**, with good performances for both stochastic and deterministic-like textures. This model has been explored for decades, [Kas80] introducing the idea of using RFs for image modeling and [MH80] the use of MRFs for texture modeling. These ideas were further explored in [CJ83] that investigates the representation capabilities of MRFs for the synthesis of microtextures, regular textures and inhomogeneous textures. Furthermore, [CK85] shows that non-causal Auto Regressive (AR) models often have the oscillatory behavior characteristic to textures and that they can be successfully used for texture synthesis, and [CC85] provides two feature extraction methods based on the Gaussian MRF (GMRF) model for texture classification. [LB06] presents a texture synthesis algorithm based on Markov mesh models, a special case of MRF. Recent works such as [CC08], based on the AR model for image segmentation and texture classification, also illustrate the interest of this class of methods, which are rather simple, with a reduced number of parameters and easy to handle.

Valuable insight into the use of **Gibbs RFs** for texture modeling is provided by [Gim99], along with the theoretical connections between the MRFs and the Gibbs RFs, in a local versus global expression of image energy. The Gibbs model is defined based on the clique potentials, and these can be tailored in such a manner as to model multiple translation invariant pairwise pixel interactions, by making each pixel appear simultaneously in several cliques. Moreover, this extensive analysis covers the difference between Gibbs-Markov fields and Gibbs-non-Markov fields. From an algorithmic point of view, practical aspects related to texture synthesis or learning are reviewed and put into context, such as simulated annealing, the Maximum Likelihood Estimator (MLE), which are then used in applications of texture synthesis, classification, both in the case of synthetic and natural textures.

In the same spirit of RFs, [BS96] solves the problem of image segmentation using a multiscale RF model for the image. The very recent [Bou13] provides an overview of model based image processing, centered on stochastic models, either causal Gaussian, such as the AR model, or non-causal, such as the GMRF, and their comparison. Furthermore, it explores the non-Gaussian MRFs and their connection to the Gibbs RF, and also the various types of clique potentials  $\rho(\Delta)$ , either convex:

- Gaussian :  $|\Delta|^2/2$ ,
- Total Variation :  $|\Delta|$ ,
- Generalized GMRF :  $|\Delta|^p/p$ ,
- Huber : 
$$\begin{cases} \Delta^2/2 & \text{for } |\Delta| < T \\ T|\Delta| - T^2/2 & \text{for } |\Delta| > T, \end{cases}$$

or non-convex

- Blake-Zisserman :  $\min \{|\Delta|^2, T^2\}$ ,
- Geman-McClure :  $\frac{\Delta^2}{\Delta^2 + T^2}$ ,
- Hebert-Leahy :  $\log(\Delta^2 + T^2)$ ,
- Geman-Reynolds :  $\frac{|\Delta|}{|\Delta| + T}$ ,

Furthermore, the Ising model is also presented, as a particular case of a **discrete MRF**. Moreover, this work provides valuable information regarding the algorithmic implementation of stochastic sampling, *i.e.*, implementation options for the described models.

A series of works devoted to finding accurate representations for "natural/photographic" images have also been carried out in the wavelet domain and tried to determine adequate decompositions and models for the wavelet coefficients. For instance, [WSW01] defines random cascades on trees of multiresolution coefficients, these cascades reproducing a **Gaussian Scale Mixtures** (GSM), a special class of random variables. This model is also presented in [Sim05] in an overview of statistical modeling techniques. The GSM wavelet coefficients image model is further used in an image denoising application [PSWS03] and refined in [HS08], to deal with orientations, and in [LS09], for multiscale subband modeling. Although not specifically devoted to texture representation, this model successfully captures the characteristics of natural images, which in most cases contain texture. The GSM modeling in the wavelet domain has represented one of the inspirations for our non-Gaussian RF based texture model that will be presented in the following section.

\*\*\*

Far from being exhaustive, this section has presented a brief review of the main branches of approaches to texture representation and reference papers in the texture related literature. More or less present in the literature, each of these models has its advantages and its own set of texture classes and applications for which it outperforms from a certain viewpoint the concurrent approaches. Consequently, the question of which is the most appropriate texture model is not trivial and does not have an unique answer.

The intention of embedding the texture model into a complex image processing application has definitely biased our choice. Firstly, we have focused our attention towards models that have both analysis and synthesis capabilities and, secondly, towards highly structured models that manage to capture the texture specificities in a relatively reduced set of parameters.

## 3.2 Texture Modeling by Random Fields

The statistical approach towards the addressed inverse problems has only made it coherent that a statistical model should also be employed for the texture. Moreover, the previous section has presented a part of the work carried in this domain, which confirms the extended representation capabilities of this type of models, both in texture analysis and in texture synthesis tasks.

Our attention has focused on indirect observations, *i.e.*, textures corrupted by a blur and by noise. This convolution based observation system, under the circulant assumptions detailed in Chapter 2, can be expressed as a simple multiplication and addition in the Fourier domain. This has inspired us into defining a texture model directly in the Fourier domain. This model is able to perform both analysis and synthesis tasks, as opposed to the majority of the works based on this representation domain, which are mostly focused on extracting relevant features for texture analysis.

The main idea is to consider that the Fourier coefficients are modeled by an RF and to assign them a probability distribution.

**Remark:** A very important assumption is that the textures are stationary.

The direct implication of this assumption is that the covariance matrix has a Toeplitz-block-Toeplitz structure. Moreover, by Whittle approximation [P.W], this matrix has a Circulant-block-Circulant form, thus is diagonalizable by Fourier transform. Consequently, the Fourier coefficients,  $\hat{x}_p$ ,  $p = 1 \dots P$ , are decorrelated.

**Remark:** Furthermore, in the case where the RF is Gaussian, decorrelation and independence are equivalent, thus the Fourier coefficients are actually independent.

From an algorithmic point of view, the independence of the Fourier coefficients implies computational efficiency, since all the computations can be done in parallel for all the coefficients. Nevertheless, from a modeling viewpoint, this represents a limitation.

The Gaussian RF (GRF) model is fully described by the second order statistics of the coefficients. The literature on texture modeling records a series of works carried in the field of human perception, which tried to establish the perception sensitivity to the higher order statistics. The first conjecture has been formulated in the early work [Jul62] and stated that humans were unable to preattentively distinguish between textures having the same second order statistics and different higher order statistics, called isodipoles. However, this conjecture has been proven wrong, by Julesz himself and a series of other researchers. Preattentively distinguishable isodipoles and even isotrinomes (textures with identical third order statistics and different higher order statistics) have been synthesized [JGSF73, Jul80, Vic94, MN01], see Figure 3.1 for an example. Nevertheless, all these textures share a common characteristic, the fact that they are very structured, deterministic-like.

The conclusion that can be drawn from this extensive work is that, although not capable of describing all texture classes, the second order statistics encode however a great amount of information regarding textural content. This shows that the GRF model, despite its limitations, is an interesting tool in texture processing.

In this work we are only dealing with stationary textures, consequently, we will not be interested in the phase field [Bro99]. Thus, the useful information will be encoded in the modulus, and the phase field may differ among realizations of textures with the same modulus field. In order to provide a simple modeling we chose to impose constraints directly on the PSD, without separately controlling the real and the imaginary components.

In the following we will present two texture models based on GRFs:

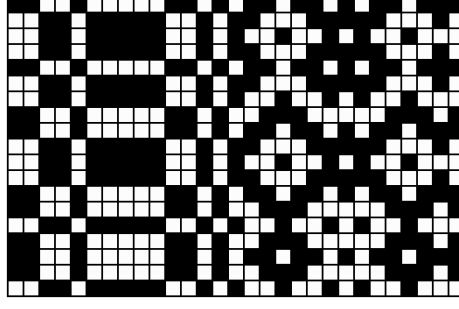


Figure 3.1: Example of an isopole pair, presented in [Jul80].

1. a model based on a Gaussian law for the texture Fourier coefficients,
2. a model based on a non-Gaussian law for the texture Fourier coefficients, with enhanced representation capabilities due to the thicker tails of the law.

The models are fully specified by the mean and the PSD, or by the mean, PSD and auxiliary variables law, respectively. In this work we will be treating the zero-mean case. Both of these GRF based models consist in formulating a Gaussian law for the Fourier coefficients of the image, conditionally on the PSD.

Then, the conditional law for the image in the spatial domain can be written as:

$$f(\mathbf{x}|\mathbf{R}_x) = (2\pi)^{-P} |\mathbf{R}_x|^{-1} \cdot \exp \left[ -\|\mathbf{x}\|_{\mathbf{R}_x}^2 \right] \quad (3.1)$$

where  $\mathbf{R}_x$  is the Circulant-block-Circulant covariance matrix. Due to this particular structure, the Fourier coefficients are independent and thus the previous expression has a separable counterpart in the frequency domain:

$$f(\mathring{\mathbf{x}}|\boldsymbol{\chi}) = (2\pi)^{-P} \prod_{p=1}^P \chi_p \cdot \exp \left[ -\sum_{p=1}^P \chi_p |\mathring{x}_p|^2 \right] \quad (3.2)$$

where  $\chi_p$  represents the element of the inverse PSD at position  $p$ . More specifically,  $\chi_p = s_p \lambda_p$  is the product of two components:

- $s_p$  is the scale parameter,
- $\lambda_p$  is the shape component.

In other words,

$$\mathring{x}_p | \chi_p \sim \mathcal{N}(0, \chi_p^{-1}) \quad (3.3)$$

*i.e.*, the Fourier coefficients are independent and non-identically distributed (inid).

### 3.2.1 Gaussian Model

The GRF texture model is very simple, while at the same time capable of representing rather complex textural contents. This is in fact the exact model employed in our work on fast texture parameters sampling [VGB11], and on pixel interaction model choice [VGR12]. In the following, we will present this model in detail, along with typical texture realizations.

The particularity of this model is that all the scale parameters are equal:  $s_p = s$ , for  $p = 1 \dots P$ . Then, (3.2) becomes:

$$f(\hat{\mathbf{x}}|s, \boldsymbol{\lambda}) = (2\pi)^{-P} s^P \prod_{p=1}^P \lambda_p \cdot \exp \left[ -s \sum_{p=1}^P \lambda_p |\hat{x}_p|^2 \right] \quad (3.4)$$

*i.e.*, the conditional law of the Fourier coefficients differs only in the shape component of the PSD,  $\lambda_p$ :

$$\hat{x}_p|s, \lambda_p \sim \mathcal{N} \left( 0, (s\lambda_p)^{-1} \right). \quad (3.5)$$

In order to fully specify the model, the prior for  $s$  must also be set. For conjugacy reasons, the chosen law has a Gamma form. Then, the GRF texture model can be hierarchically represented as:

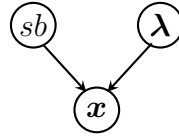


Figure 3.2: GRF texture model

A texture realization can be obtained using virtually any form for the PSD, from the simplest uniform texture, obtained from a PSD with a single impulsional component at the null-frequency, to a random noise, corresponding to a uniform PSD. The complexity of the texture is directly related to the richness and the structure of its PSD.

Figure 3.3 shows a series of texture realizations with some of the simplest possible PSDs. We have chosen combinations of one, two, four and nine pure frequency components, respectively. The goal is to illustrate the type of patterns that can be obtained with such a reduced number of components. Obviously, the number and position of these frequency components determine the characteristics of the corresponding texture. Consequently, a richer spectral content would provide more complex patterns. Moreover, the goal is to find an efficient manner to encode this richer spectral content and one option is the use of parametric models.

The idea of using parametric models for the PSD is very practical, since textures with more complex spectral contents can be obtained using a reduced number of parameters. Figure 3.4 shows a series of parametric PSD shapes, along with the corresponding texture realizations. Notice the effect of changing the position of the component, on the textures in Figures 3.4a and 3.4b, and the effect of changing the parametric shape, between the textures in Figures 3.4a and 3.4c.

In these cases, the models depend on at most four or five parameters, which, along with the model for the PSD shape, fully encode the information regarding the texture. As it can be seen in the presented realizations, these textures are more complex and resemble natural textures. The textures obtained using such a continuous PSD are very stochastic-like, with patterns where it is hard to identify a structural element.

Nevertheless, more structured, deterministic-like textures can also be obtained using this model. This can be achieved starting from the previously mentioned continuous forms,



by selecting only a part of the frequency components. Such textures are shown in Figure 3.5, being obtained using a uniform sampling lattice over the entire frequency domain. To illustrate the effect of this sampling, the texture in Figure 3.5c is obtained from the same PSD as the texture in Figure 3.4e, the difference between the two realizations being significant. Furthermore, the continuous PSD shapes used to obtain the last two textures consist in the sum of two Lorentzian components and two Laplacians, respectively.

These uniformly sampled PSDs are also parametric and, as compared to their continuous counterparts, only contain two supplementary parameters indicating the sampling

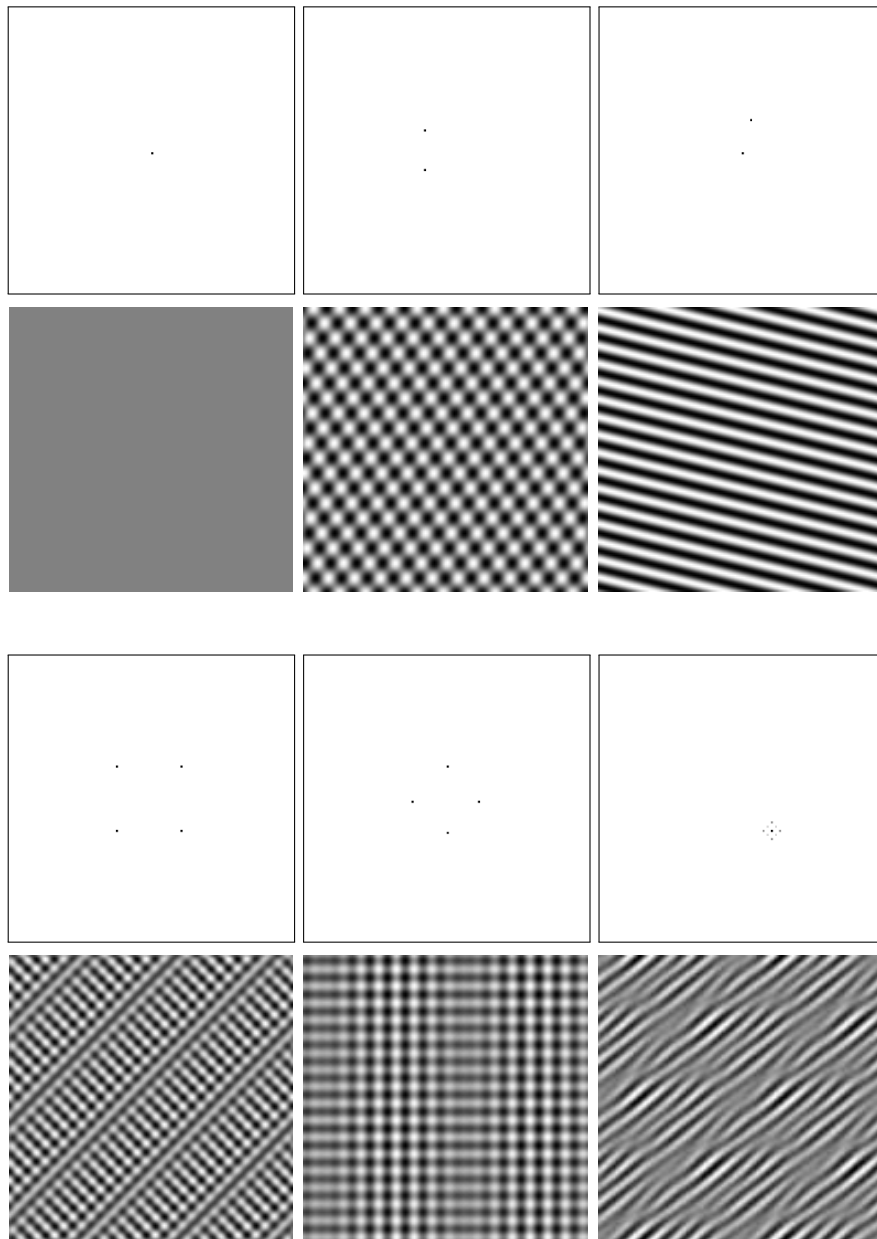


Figure 3.3: Texture realizations for the zero-mean GRF model. On the first and third rows are illustrated various forms for the PSD and underneath each PSD, the corresponding texture. The spectral content of these textures is very reduced, consisting in a small number of punctual frequency components (impulse like PSD).

frequencies on the two axes. Moreover, more complex sampling lattices can be used, lattices that can themselves be parametric.

**Remark:** These textures have a more obvious geometrical pattern, with a structural element that repeats itself, this giving them a more deterministic-like character.

The textures we have presented cover only a small part of the large variety of texture classes that can be modeled using GRFs and second order statistics. However, they show that rather complex textural content can be encoded in a reduced number of parameters and motivate the interest we have taken in this model.

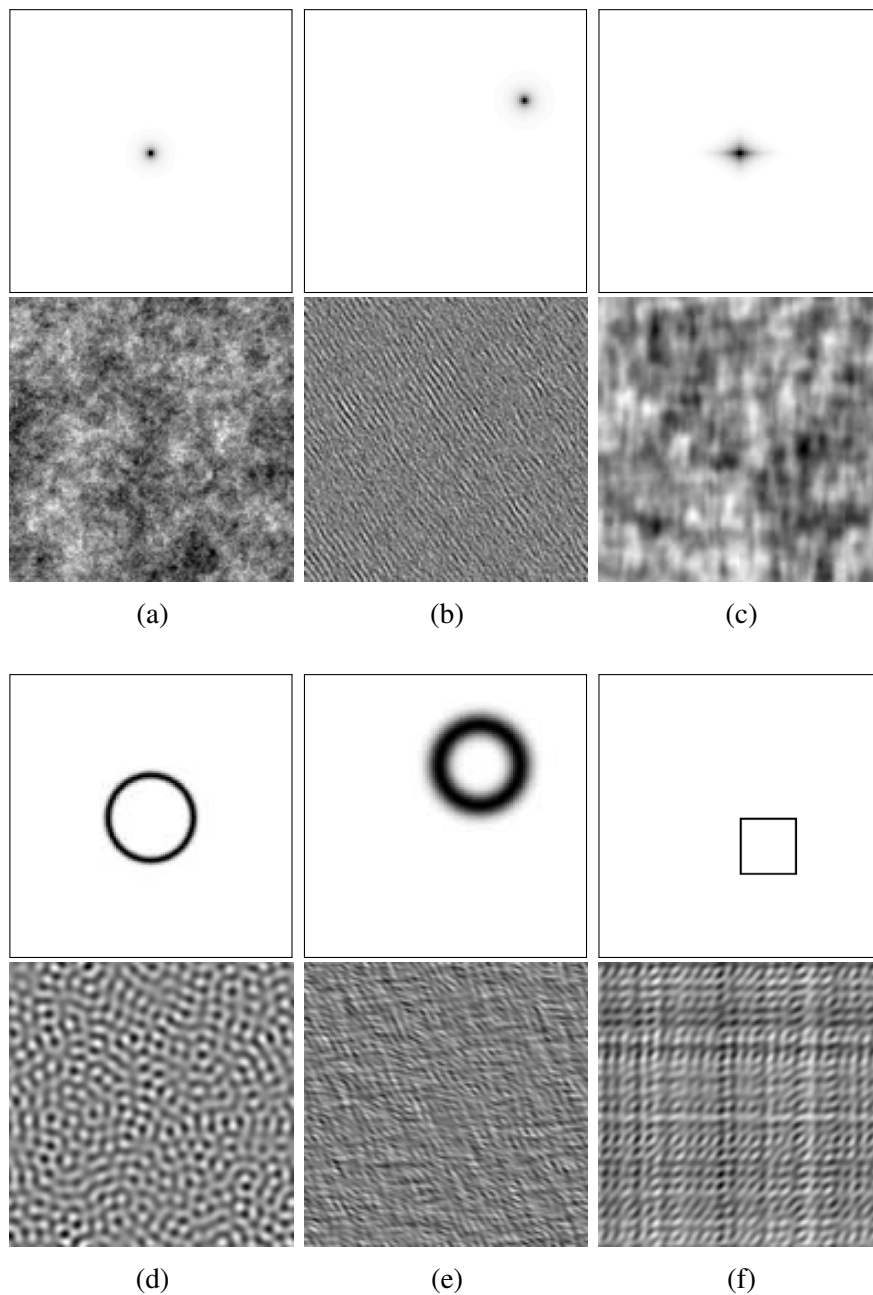


Figure 3.4: Texture realizations of a zero-mean GRF, with the PSD consisting in a single, continuous parametric shape.



The  $\lambda_p$  from Equations (3.2) and (3.4) are the elements of the PSD field and depend on the model,  $\mathcal{M} = k$ , and on the parameters of that model,  $\theta_k$ . From this point forward, to explicitly show this dependency the notation  $\lambda_p^k(\theta_k)$  is used.

Although the use of the aforementioned parametric PSDs means that the entire information regarding the textural content is encoded in a small number of parameters, the problem does not become trivial. The texture parameters will not be easy to estimate, due to the non-linearity of the data with respect to the parameters. To be more exact,  $\lambda_p^k(\theta_k)$  has a highly non-linear, non-standard form.

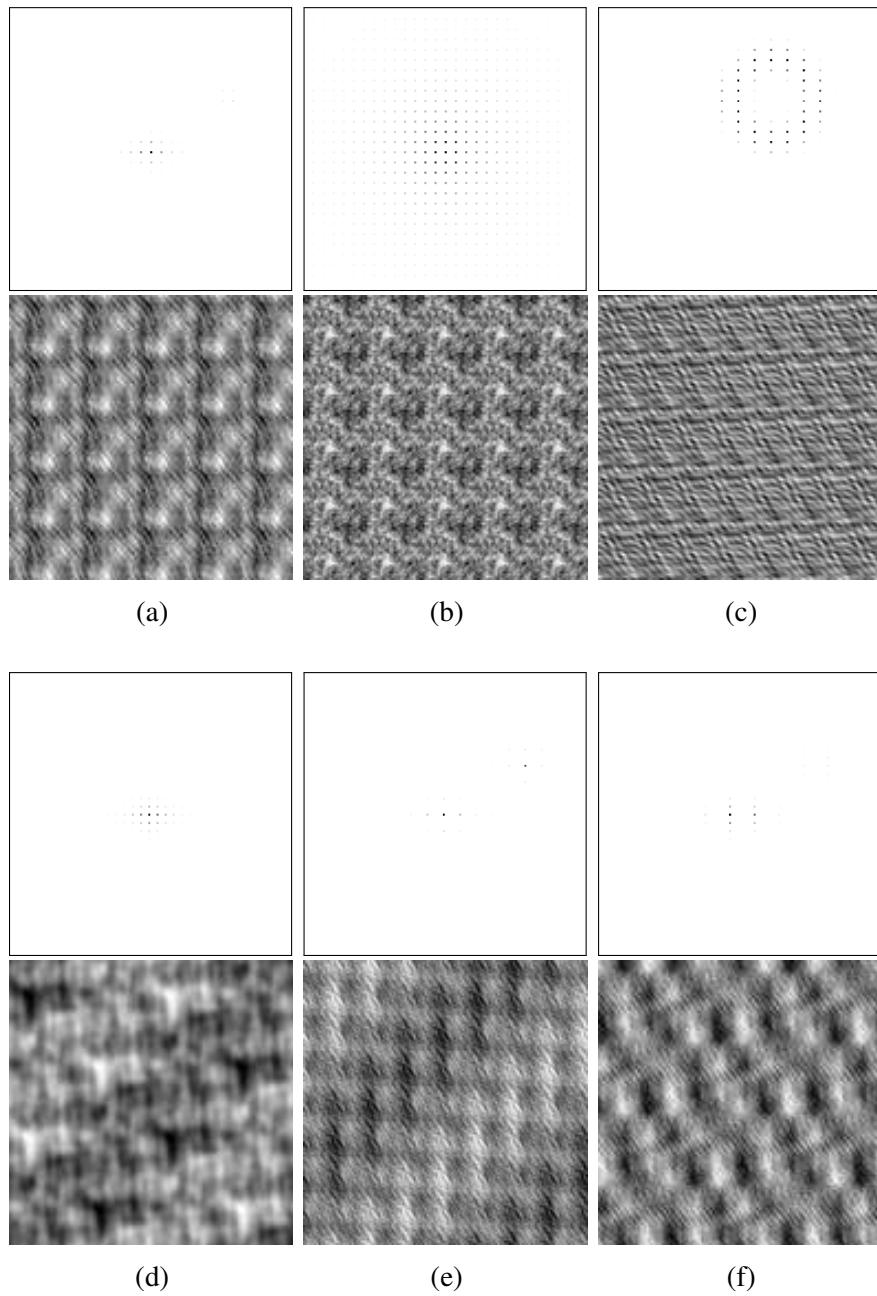


Figure 3.5: Texture realizations of a zero-mean GRF, with the PSD composed of different uniformly sampled continuous parametric shapes.

Nevertheless, this model has its limitations. Among these limitations, the Gaussianity of the Fourier coefficients can be overcome almost effortlessly. This will be represented in the following section as our non-Gaussian texture model.

### 3.2.2 Non-Gaussian Model

The previously presented Gaussian model has the main advantage that the texture Fourier coefficients are independent and follow Gaussian laws. This is extremely advantageous from a practical point of view, since it allows for parallel processing, of standard laws, on the entire texture. The idea behind the new texture model is to obtain non-Gaussian distributed Fourier coefficients, while keeping the aforementioned advantages.

This aspect has already been widely explored in statistics, but also in image processing [GY95, GR92, Gio08] to define the regularization terms in problems of deconvolution and denoising. In such problems, the data adequacy term is Gaussian with respect to the object of interest (the original image) and the goal is to conserve the overall Gaussianity, for computational efficiency. At the same time, the aim is to eliminate the limitations that a Gaussian regularization term imposes, all this being achieved by the introduction of a set of auxiliary variables. This model is called the Gaussian Scale Mixture (GSM).

A similar principle has been presented in [Ayk98] with Markov GRFs for image modeling. Furthermore, extensive work [WSW01, PSWS03, Sim05, HS08, LS09] has been carried for finding accurate models for natural images. The approach explored in this case relies on modeling the coefficients of the wavelet decomposition using GSMs. Although the idea of conditional Gaussianity and marginal non-Gaussianity has been previously used, to the best of our knowledge, it has not been exploited so far for texture modeling.

For our new texture model, the set of auxiliary variables is represented by the scale parameters  $s$ , so that the conditional law  $f(x|s, \lambda)$  is Gaussian, but the marginal law  $f(x|\lambda)$  is no longer Gaussian. In this case, the set  $s$  contains one variable per Fourier coefficient. Let this model be called a Gaussian Scale Mixture Random Field (GSMRF).

As previously stated, our focus is on stationary textures, which, under a Whittle approximation, have decorrelated Fourier coefficients. For the GSMRF model, due to the Gaussianity of the conditionals, these coefficients are independent conditionally on the scale parameters. At this point, there are two modeling alternatives, depending on the type of prior used for the scale parameters:

- independent scale parameters – as in the case of the GRF model, the Fourier coefficients will be both conditionally and marginally independent. Although this reduces the representation capabilities of the model, it renders it very computationally efficient.
- interdependent scale parameters – the texture coefficients are no longer marginally independent. Since the number of scale parameters is equal to the number of pixels, the use of an interdependent set could imply a prohibitive processing cost for the synthesis of such a texture. In this case, although the texture coefficients can be sampled in parallel, conditionally on the scale parameters, the scale parameters themselves must be sampled sequentially in a costly process.

### 3.2.2.1 GSMRF with independent scale parameters

This is the simplest modeling alternative and ensures the maximum model tractability.

**Remark:** The terminology of marginal law as opposed to conditional law, in the context presented here, only refers to the dependency with respect to the scale parameters  $s_p$ . All the laws are however conditional with respect to the shape components  $\lambda_p$ . Thus, the Fourier coefficient  $\hat{x}_p$  has the conditional law  $f(\hat{x}_p|s_p, \lambda_p)$  and the marginal law  $f(\hat{x}_p|\lambda_p)$ .

In fact, between the two texture models only the form of  $\chi_p$  changes:

- GSMRF –  $\chi_p = s_p \lambda_p$
- GRF –  $\chi_p = s \lambda_p$

This difference can be induced through the pdf of the scale parameters. Consequently, the form of the prior for  $s$  will make the difference between the GRF and the GSMRF models and through these parameters we will be able to switch between the two laws, without changing the PSD.

For the same conjugacy considerations evoked in the previous section, the prior distribution of each  $s_p$  is chosen to have a Gamma form. Then, the  $s_p$  are iid:

$$f(s_p|\alpha_s, \beta_s) \propto s_p^{\alpha_s-1} \exp[-\beta_s s_p] \quad (3.6)$$

Based on the previously defined prior for the scale parameters, the marginal law  $f(\mathbf{x}|\boldsymbol{\lambda})$  has a Student's t form:

$$\begin{aligned} f(\hat{x}_p|\lambda_p, \alpha_s, \beta_s) &= \int_{s_p} f(\hat{x}_p|s_p, \lambda_p) \cdot \pi(s_p|\alpha_s, \beta_s) ds_p \\ &\propto \left[ 1 + \frac{\lambda_p}{\beta_s} |\hat{x}_p|^2 \right]^{-\alpha_s-1} \end{aligned} \quad (3.7)$$

Moreover, when comparing the GRF and the GSMRF models, we considered it useful to be able to switch between models without inducing other changes except for the probability distribution of the coefficients. In order to evaluate the influence of this change of law on the texture characteristics, we can impose the condition of having identical second order moments for both cases, *i.e.*, having the same marginal variance of the Fourier coefficients. This allows us to make one change at a time: for instance, a change of the coefficients pdf will not affect their variance, and, conversely, a change in the PSD will not change the type of pdf. Thus, for the two types of laws, we want to have the same variance for the Fourier coefficients, conditionally on the  $\lambda_p$ :

$$\begin{aligned} \text{var}_{\text{GSMRF}} [\hat{x}_p|\lambda_p] &= \frac{\beta_s}{2\alpha_s - 1} \cdot \frac{1}{\lambda_p} \\ \text{var}_{\text{GRF}} [\hat{x}_p|\lambda_p] &= \frac{1}{\lambda_p} \end{aligned}$$

this imposing a constraint on the parameters of the prior for  $s_p$ :  $\beta_s = 2\alpha_s - 1$ , with  $\alpha_s > 0.5$ . In this manner, the two types of textures will have the same PSD, but different laws.

To summarize, in this setting, the texture synthesis will be a relatively effortless process, consisting in two sampling stages:

1. the  $s_p$ : independent, with Gamma pdf,
2. the  $\hat{x}_p|s_p$ : independent, with Gaussian pdf.

Realizations of the GRF and GSMRF models, with the same PSDs, is given in Figure 3.6. On the first column there are realizations of the GRF texture model with a certain PSD, while on the second column are the realizations of the corresponding GSMRF models, with the same PSDs. Due to their common features:

- independent Fourier coefficients,
- second order statistics,
- the same PSD,

the GRF and the GSMRF models yield rather similar, primarily stochastic textures. However, the GSMRF is able to generate slightly more complex patterns. This is due to the fact that, although both models have the same PSD, the equality holds only under the expectation. For one sample, the frequency content of the GSMRF textures will be more complex due to the scale parameters.

### 3.2.2.2 GSMRF with dependent auxiliary variables

The choice of using independent scale parameters is a limitation for the model and it is meant to ensure that the model is tractable and the texture synthesis does not become prohibitively expensive. There are however options for defining a certain dependence among the scale parameters without dramatically increasing the complexity.

One of these options is to model the scale parameters by a Potts field (see Appendix C for a description of this model). If the cliques are limited on the 4-neighborhood, the  $s$  field can be sampled in only two steps, by sampling in parallel all the variables not being neighbors. If the size of the neighborhood increases, so will the number of steps.

However, the use of such a prior for the auxiliary variables may not bring a considerable change for the texture characteristics. This is illustrated by Figure 3.7 where we show two texture realizations of the GSMRF model, the first corresponding to a Potts prior for the scale parameters and the second to a Gamma separable prior. Surprisingly, although the scale parameters fields are significantly different, the corresponding periodograms are very similar. This is reflected in the textures themselves, which are practically indistinguishable.

In the light of this observation, the use of the Gamma prior for the scale parameters is not only justified by its ease of implementation, but also by the fact that more complicated, non-separable models might not bring any gain in terms of representation capability.

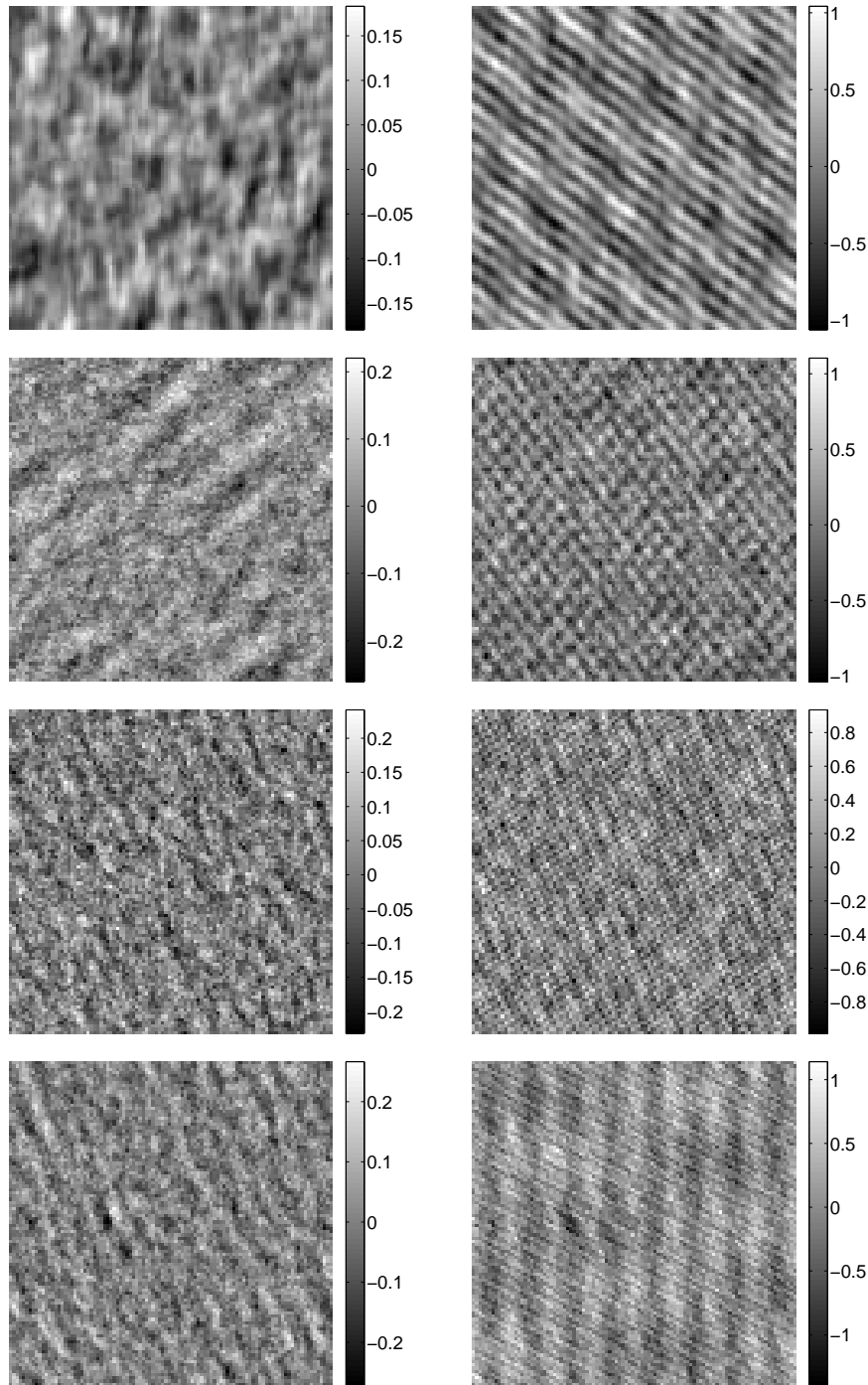


Figure 3.6: Texture realizations for the GRF (left) and the GSMRF (right) models with the same PSD.

### 3.3 Conclusion and Perspectives

This chapter has presented the mathematical formulation and a series of realizations of our SMGRF texture model. This rather simple model is formulated in the Fourier domain and relies on the principle of conditional Gaussianity with respect to a set of auxiliary variables which renders the marginal law non-Gaussian. Due to this property, this model

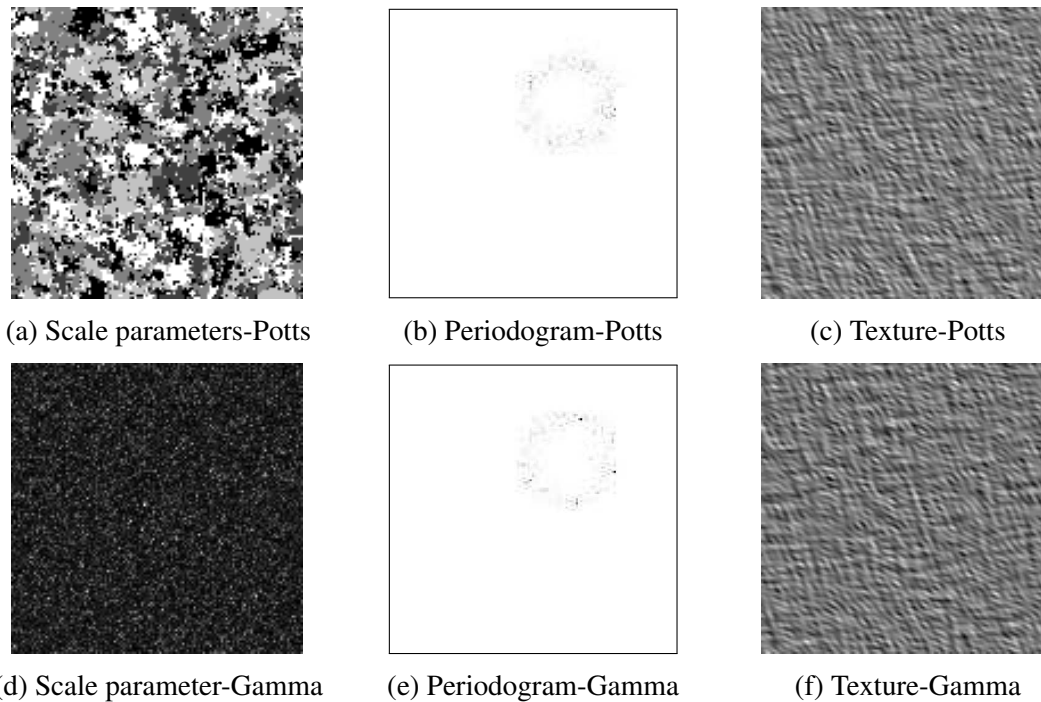


Figure 3.7: Texture realizations for the GSMRF model with scale parameters having a Potts prior and a Gamma prior, respectively.

is very tractable, which makes it fit the constraints of the inverse problems in which it will be integrated.

Despite the series of choices that have been done to achieve this easy to handle form, the model has good representation capabilities, especially for stochastic textures. Moreover, some of these limitations may not even have a strong impact on the model's representation capabilities, as shown in the previous section, where the use of a non-separable prior for the scale parameters did not enhance the representation capabilities of the model and the texture realizations were indistinguishable from those obtained using the separable prior.

Nevertheless, more complex priors can be explored and this is the main perspective of this work on texture modeling.

# Unsupervised Myopic Deconvolution of a Textured Image

## Contents

<b>4.1 Problem Statement</b>	<b>37</b>
4.1.1 Deconvolution	39
4.1.2 Data, image and TF models	41
<b>4.2 Myopic deconvolution for textured images</b>	<b>42</b>
4.2.1 Information and qualitative estimation performance analysis	42
4.2.2 Bayesian setting: priors, posterior and conditional posteriors	46
<b>4.3 Results</b>	<b>48</b>
<b>4.4 Conclusion and Perspectives</b>	<b>51</b>

Chapter 2 has presented a complex inversion paradigm where the data  $\mathbf{y}$  is a blurred and noisy version of the original image. This original image is made up of  $K$  regions, each region containing a different texture  $\mathbf{x}_k$ ,  $k = 1 \dots K$ . The hidden label variables  $\mathbf{z}$  indicate for each pixel the region to which it belongs.

In Chapter 3 we have provided a detailed description of our GRF based texture models and their properties, which make them very suited to be used in the inversion problem. Consequently, we consider that the original images  $\mathbf{x}_k$  are modeled by GRFs.

The problem becomes estimating the original textured images, determining the underlying model  $\mathcal{M}_k$  and its parameters for each image and the hidden label variables  $\mathbf{z}$  in an unsupervised manner. Due to the large number of the unknowns and the multiple layers of difficulty, we have decided to divide the problem into a series of subproblems of lower complexity. Notwithstanding, even these subproblems have a high level of complexity and, moreover, are of significant interest since they address important topics in image processing.

## 4.1 Problem Statement

One of the subproblems that can be considered is the unsupervised myopic deconvolution of a textured image with PSD parameter estimation. The term **unsupervised** refers to the fact that the hyperparameters  $\gamma_n$  (noise precision) and  $\gamma_x$  (global scale parameter for the PSD of the textured image) are also estimated, while the term **myopic** indicates that the

PSF has a parametric form. This topic has made the subject of our publication [VGB14]. The particularities of this problem are the following:

- original image  $x$  consisting in a single texture. The texture is modeled by a GRF (no auxiliary variables  $s$ ), with known PSD model, driven by the unknown parameters  $\theta$ ,
- the PSF has a known parametric model, driven by the unknown parameters  $\eta$ .

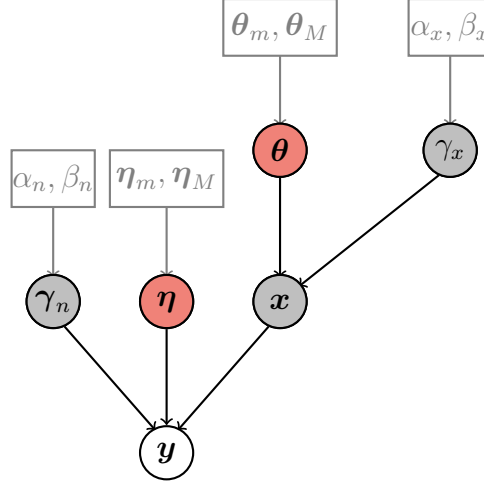


Figure 4.1: Hierarchical variable dependencies for the textured image unsupervised myopic deconvolution.

In this setting, the dependencies between the data and the unknowns can be graphically represented as shown by the graph in Figure 4.1.

The goal is to estimate the PSD and PSF parameters,  $\theta$  and  $\eta$ , respectively. We have devised an unsupervised method that provides optimal estimates for these parameters. Therefore, our method provides, in addition to the original textured image, estimates for the:

1. texture parameters,  $\theta$ ,
2. instrument parameters,  $\eta$ ,
3. signal and noise levels,  $\gamma_x, \gamma_n$ .

This distinguishes our method from the existing works. These methods also perform image deconvolution and denoising by formulating regularization terms and estimating the noise parameter, however, they do not use parametric regularization terms.

The problem formalization, in the context of the textured images presented in Chapter 3, can be represented as:



where  $x_{(\theta, \gamma_x)}$  is the textured image and  $H_\eta$  is the parametric PSF, having a circulant-block



circulant structure, due to the Whittle approximation detailed in Chapter 2. Let us denote by  $\Psi = \{\theta, \eta, \gamma_x, \gamma_n\}$  the full set of unknowns.

The observation model is convolutional and can be described by the equation:

$$\mathbf{y} = \mathbf{H}_\eta \cdot \mathbf{x}_{(\gamma_x, \theta)} + \mathbf{n}_{(\gamma_n)} \quad (4.1)$$

$\mathbf{x}$  and  $\mathbf{n}$  are modeled by zero-mean, stationary GRFs, independent, of covariance matrices  $\mathbf{R}_x(\theta)$  and  $\mathbf{R}_n(\gamma_n)$ . From the fields stationarity and the Whittle approximation, these covariance matrices are circulant-block circulant and are thus diagonalizable by Discrete Fourier transform (DFT). Hence, the Fourier coefficients of the data are independent:

$$\hat{y}_p = \hat{h}_p \cdot \hat{x}_p + \hat{n}_p \quad (4.2)$$

and can be computed by Fast Fourier Transform (FFT).

In order to fully exploit the laws separability in the mathematical developments, the PSF, the noise and texture PSDs are written in the frequency domain. From this point forward we will be dealing with the frequency domain counterpart of the PSF, the Transfer Function (TF).

### 4.1.1 Deconvolution

The literature on image deconvolution is vast and covers various regularization forms and various representation domains for the image, either in a deterministic or in a probabilistic approach. Furthermore, from the PSF model point of view, there are two concurrent approaches to the deconvolution problem, each with its advantages and its weaknesses.

#### 4.1.1.1 Blind deconvolution

Blind deconvolution is the classic approach to image deconvolution, most of the existing works in the literature being devoted to this type of modeling. It consists in considering the PSF as having a non-parametric model, represented by an impulse response matrix  $\mathbf{H}$  with unknown elements. Since the convolution is a linear operation, the dependency of the data with respect to the elements of this matrix is also linear. Hence, the estimation process in this context consists in determining the PSF elements, from a linear system of equations.

The problem is still ill-posed, since the number of unknowns is higher than the number of observations and consequently regularization is used in order to achieve a unique solution. In this framework, the regularization is done on the image and on the PSF and is based on the idea that in most cases, the images are piecewise smooth, with a reduced number of discontinuities (contours). Thusly, an adapted regularization is to impose a low frequency model with sparse transitions for the image and a low frequency model for the PSF. This is achieved by imposing various constraints through either stochastic models such as Simultaneous Auto-Regressions, Student's t, etc., or deterministic penalties such as L1 norm, L2-L1 norm, L2 norm, Total Variation. Such regularization forms for the

image are limited in terms of representation capability and make the methods appropriate mostly for dealing with piecewise-smooth or piecewise-constant images.

The regularization terms can be formulated in a fully deterministic manner, as is the case in [FBD10, BDF10, ABDF11], where the solution is determined by convex optimization. The alternative approach is the Bayesian formulation, used in works such as [MMK06, BMK09, BWMK10, TLG07, TLG09], where the aforementioned regularization forms are introduced as priors on the image and the PSF. In both of these cases, the image regularization can be formulated in various representation domains, the vast majority of the literature being devoted either to the spatial representation [MMK06, BMK09, BWMK10] or to the wavelet domain [VU08, TLG07, TLG09, FBD10, BDF10, ABDF11].

The aforementioned methods all have a high degree of complexity and do not allow for an explicit analytical solution. For this reason, numerical methods must be employed in order to solve these problems. Examples of such methods are for instance the algorithms of Alternate Method of Multipliers for the deterministic cases, or the Variational Bayes for the probabilistic approaches.

The advantage of the blind deconvolution method consists in the linearity of the criterion with respect to the elements of the PSF. This meaning that conjugate priors can be employed and, consequently, that the posterior law for these elements has a standard form. [CE07] offers a detailed review of the blind deconvolution techniques and their applications.

#### 4.1.1.2 Myopic deconvolution

Myopic, or semi-blind, deconvolution is the alternative to the blind approach. It is based on a parametric model for the PSF, driven by a set of parameters of reduced size. In practice, information regarding the PSF is often available, especially about its form. Moreover, it is not uncommon for this form to be parametric.

The non-linearity of the PSF with respect to its coefficients is for instance typical in astronomy. The problem of myopic deconvolution for astronomical images is tackled in [OGR10] in a Bayesian framework using an L2-L1 regularization on the image.

In another line of work, [PDH12, PDH14] address the problem of myopic deconvolution from a different perspective, by overcoming the non-linearity of the PDF through its decomposition onto a basis.

At a first glance, myopic deconvolution might seem like a simpler problem as compared to the blind deconvolution case. Indeed, the number of unknowns becomes smaller, however, usually, the form of the dependency with respect to the parameters of the PSF is no longer linear and in most cases is rather complicated. Consequently, there will be no conjugate priors to enable a straightforward sampling procedure. In such a situation, adapted sampling algorithms must be employed, the downside being that this can be rather costly.

In other words, the main difference between the blind and myopic approaches is the manner in which they constrain an ill-posed problem. For the blind case, the supplementary information needed to constrain the solution is represented by the regularization term over

the PSF coefficients, while for the myopic case the parametric model itself is strongly structuring.

### 4.1.2 Data, image and TF models

The previous section has presented the two approaches for image deconvolution, blind and myopic. In the following we will present our method based on the latter approach, we will provide the models we have used and the corresponding mathematical developments.

From the noise law and (4.2) follows the relation for the Fourier coefficients of the observations  $\mathring{y}$ , texture  $\mathring{x}$ , filter's TF  $\mathring{h}$  and hyperparameters  $\gamma_x, \gamma_n$ :

$$f(\mathring{y}_p | \mathring{x}_p, \boldsymbol{\eta}, \gamma_n) \propto \mu_p(\gamma_n) \cdot \exp \left[ -\mu_p(\gamma_n) \left| \mathring{y}_p - \mathring{h}_p(\boldsymbol{\eta}) \mathring{x}_p \right|^2 \right] \quad (4.3)$$

$$f(\mathring{x}_p | \boldsymbol{\theta}, \gamma_x) \propto \gamma_x \lambda_p(\boldsymbol{\theta}) \cdot \exp \left[ -\gamma_x \lambda_p(\boldsymbol{\theta}) \left| \mathring{x}_p \right|^2 \right] \quad (4.4)$$

where  $\mu_p$  are the eigenvalues of  $\mathbf{R}_n^{-1}$ .

The  $\mathring{h}_p(\boldsymbol{\eta})$  and  $\lambda_p(\boldsymbol{\theta})$  from (4.3) and (4.4), respectively, represent the TF values and the inverse PSD, respectively, computed on the discretized frequency domain at position  $p$ .  $\boldsymbol{\lambda}(\nu_x, \nu_y, \boldsymbol{\theta})$  represents the inverse of the texture's PSD.  $(\nu_x, \nu_y)$  belongs to the reduced frequency domain and  $(\nu_n, \nu_m) = (n\Delta\nu, m\Delta\nu)$  is pixel's  $p$  location.

An interesting aspect is the fact that the method is adapted to any TF and any PSD form for the texture and the noise. However, to illustrate the mathematical developments and numerically evaluate the algorithm, we have chosen:

- TF – low pass filter (Dirichlet kernel), of width  $\eta_x = \eta_y = \eta$ :

$$\mathring{h}(\nu_x, \nu_y, \eta) = \frac{1}{2\eta} \frac{\sin(2\pi\eta\sqrt{\nu_x^2 + \nu_y^2})}{\sin(\pi\sqrt{\nu_x^2 + \nu_y^2})} \quad (4.5)$$

- noise – white noise of covariance  $R_n(\gamma_n) = \gamma_n^{-1} \mathbf{I}$ , with  $\gamma_n$  inverse variance / precision parameter.
- image – exponential model for the PSD:

$$\boldsymbol{\lambda}^{-1}(\nu_x, \nu_y, \gamma_x, \boldsymbol{\theta}) = \exp - \left[ \frac{|\nu_x - \nu_x^0|}{u_x} + \frac{|\nu_y - \nu_y^0|}{u_y} \right] \quad (4.6)$$

with  $\boldsymbol{\theta} = \{\nu_x^0, \nu_y^0, u_x, u_y\} \in \mathbb{R}^4$ .  $\nu_x^0, \nu_y^0$  are the central frequencies and  $u_x, u_y$  are the PSD widths.  $\lambda_p(\boldsymbol{\theta})$  from (4.4) is in fact obtained as  $\lambda_p(\boldsymbol{\theta}) = \boldsymbol{\lambda}(\nu_n, \nu_m, \boldsymbol{\theta})$ .

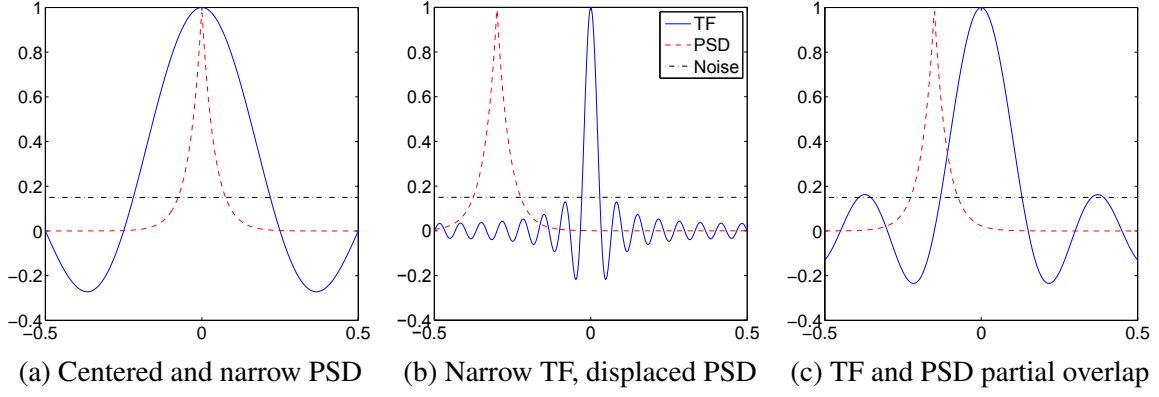


Figure 4.2: Illustration of different relative positioning and widths for the TF and PSD, resulting in different situations information-wise.

## 4.2 Myopic deconvolution for textured images

As seen in (4.5) and (4.6), the TF and PSD are driven by the unknown parameter sets  $\eta$  and  $\theta$ . The coefficients of both characteristics have a highly non-linear dependency with respect to the parameters, *i.e.*, the estimation process is rather challenging.

### 4.2.1 Information and qualitative estimation performance analysis

The parameter estimation performances are directly related to the amount of available information regarding each parameter. Consequently, the Fisher information is extremely useful in evaluating the different situations that may occur, in function of the SNR and the parameter values. Different scenarios are illustrated by Figure 4.2, as 1D cross-sections of the 2D frequency domain:

- (a) for narrow TFs (small  $\eta$ ) and high frequency PSDs (large  $(\nu_x^0, \nu_y^0)$ ), the spectral contents cancel each other, *i.e.*, the data is informative on  $\gamma_n$ , but not on  $\eta$  and  $\theta$ ;
- (b) for wide TFs and narrow PSDs, the input stimulus is incapable to induce an adequate system perturbation, *i.e.*, the information is insufficient to estimate  $\eta$ ;
- (c) ideal situation information-wise, *i.e.*, partial overlap. The information available for the estimation depends on this overlap.

Appendix A presents the analytical expressions for the Fisher information for each of the parameters. Based on these analytical forms, we can provide a detailed qualitative and quantitative analysis of the mean available information concerning each parameter of the problem.

The hyperparameters  $\gamma_n$  and  $\gamma_x$  have an important role in the estimation process since their ratio  $\gamma_n/\gamma_x$  represents the Signal to Noise Ratio (SNR). A high SNR means that the signal level is high enough so that the noise corruption cannot mask the useful information. Thus, in such a case, the estimation performances for all the parameters are high. The only exception is the noise level itself, which is better estimated when the SNR is low.

The following subsections provide detailed considerations concerning the available information about the noise parameter, a PSD central frequency and a PSD width. The variation of this quantity is taken with respect to one parameter at a time, in order to show the strong dependencies and to identify the invariant situations.

#### 4.2.1.1 Noise parameter $\gamma_n$

The analytical expression of  $\mathcal{J}(\gamma_n)$  is:

$$\mathcal{J}(\gamma_n) = \gamma_n^{-2} \sum_p \left[ 1 + \gamma_n / \gamma_x \cdot \frac{g_p(\eta)}{s_p \lambda_p(\boldsymbol{\theta})} \right]^{-2} \quad (4.7)$$

where  $g_p(\eta) = |\hat{h}_p(\eta)|^2$

Figure 4.3 illustrates the variation of the amount of mean information concerning the noise parameter. On the  $x$  axis we have represented a very wide range of possible values for the noise, since no prior information is available about this parameter. The representation is done in logarithmic scale for both axes.

The SNR sweep, which is equivalent to modifying the  $\gamma_x$ , shows that an SNR increase triggers a decrease of  $\mathcal{J}(\gamma_n)$ , *i.e.*, the estimation of the noise parameter will be more difficult. With respect to the widths of the TF and the PSD,  $\eta$  and  $u$ , respectively, the dependency is similar, that is the higher the width, the less the information on  $\gamma_n$ .

One of the most interesting considerations is related to the relative positioning of the TF and the PSD. Figure 4.3d illustrates the Fisher information for the three cases previously presented in Figure 4.2. It is interesting to notice that while the case of total overlap coincides with a lower information, there is practically no difference between the cases with no or partial overlap. This means that even in a case with no overlap,  $\gamma_n$  should be well estimated.

A more quantitative-oriented analysis can also be made. For this purpose, let us take a closer look at Figure 4.3a, for instance. For  $\gamma_n = 10^{-2}$ , the corresponding Fisher information is  $\mathcal{J}(10^{-2}) = 4 \cdot 10^7$ , while  $\mathcal{J}(10^{-1}) = 4 \cdot 10^5$ , meaning that the gain of an order of magnitude for the noise level implies the gain of two orders of magnitude in terms of information amount.

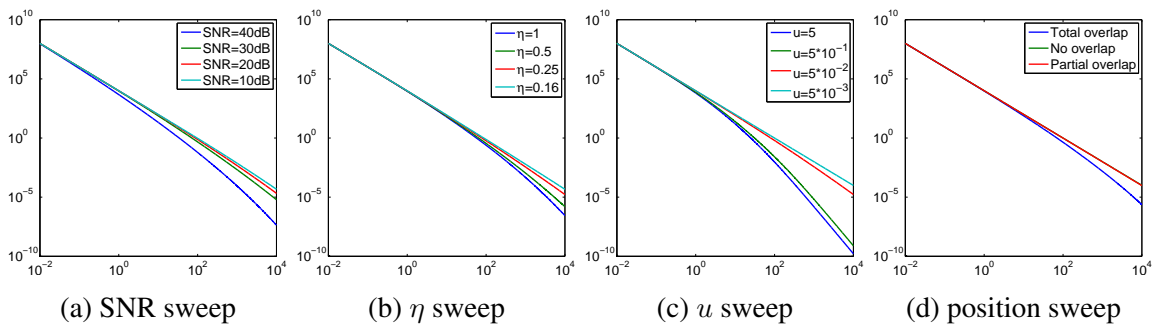


Figure 4.3: Fisher information for the noise parameter  $\gamma_n$  in the case of different signal levels, filter widths, PSD widths and relative TF and PSD positioning.

**Remark:** A general trend can be noticed in Figure 4.3 and it consists in a higher sensitivity to the parameter variation for lower noise levels (high  $\gamma_n$ ), while for the high noise levels this is almost indistinguishable. This means that in cases where the noise level is high enough, its estimation will be invariant to the rest of the factors involved.

#### 4.2.1.2 PSD central frequency $\nu_x^0$

The Fisher information for  $\nu_x^0$ , in the case of a Laplace shaped PSD, writes:

$$\mathcal{J}(\nu_x^0) = \frac{1}{u_x^2} \sum_p \left[ \frac{g_p(\eta)/\lambda_p(\boldsymbol{\theta})}{\gamma_x/\gamma_n + g_p(\eta)/\lambda_p(\boldsymbol{\theta})} \right]^2 \quad (4.8)$$

The Fisher information variation for the texture parameters  $\boldsymbol{\theta}$  is illustrated by Figure 4.4. On the  $x$  axis it is represented the range of possible values for the central frequency, *i.e.*, the reduced frequency domain. The representation is done in logarithmic scale on the  $y$  axis.

The plot representing the SNR influence clearly shows that the amount of information is strongly influenced by the SNR. Thusly, we may assert that the estimation performances should be significantly superior in the cases with higher SNR and the performance degradation when the SNR decreases should be rather significant.

Figure 4.4b shows the filter width influence and gives valuable insight for the entire information analysis. First of all, no matter the value of  $\eta$ , the Fisher information has very similar values around the null frequency. The difference intervenes at higher central frequencies. The reason for this is the presence of the blurring filter, which has a low-pass character. This means that, irrespectively of the value of  $\eta$ , if  $\nu^0$  is close to the null frequency, it will be in the filter bandpass and thus the amount of information is high. However, if  $\nu^0$  is high, the filter width becomes a factor and we notice a decrease of the information for small values of  $\eta$ . This is confirmed by the case where the filter TF is very wide, corresponding to a situation where there is almost no blurring. In this case,  $\mathcal{J}(\nu^0)$  is almost constant over the entire interval, as it can be seen in Figure 4.4b.

**Remark:** The previous considerations apply to the variation with respect to the other pa-

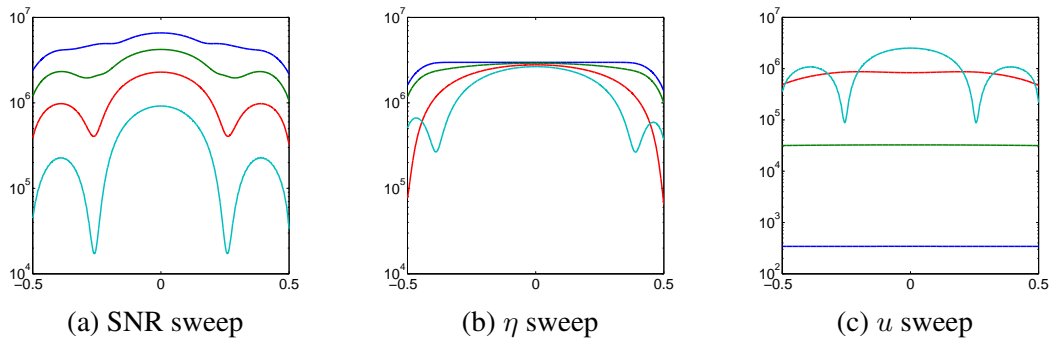


Figure 4.4: Fisher information for the PSD central frequency  $\nu_x^0$  as a function of different parameters. The legends are the same as those from Figure 4.3.

rameters as well and explain this generalized behavior with a maximum in the null frequency and local minima in the frequencies where the filter TF is zero.

The corresponding PSD width  $u_x$  also influences  $\mathcal{J}(\nu_x^0)$ , a larger width meaning larger uncertainty and thus less information about the position. Nevertheless, the width  $u_y$  has a very reduced impact on  $\mathcal{J}(\nu_x^0)$ .

The central frequency  $\nu_y^0$  intervenes in the expression of  $\mathcal{J}(\nu_x^0)$  through the  $g_p(\eta)/\lambda_p(\theta)$  ratio and its influence is due to the presence of the filter. Consequently, the dependency of  $\mathcal{J}(\nu_x^0)$  will have an oscillating character, in function of the PSD positioning with respect to the local maxima or zeros of the filter TF.

#### 4.2.1.3 PSD width $u_x$

The Fisher information for  $u_x$ , for the Laplace shaped PSD, has the following expression:

$$\mathcal{J}(u_x) = \frac{1}{u_x^4} \sum_p \left[ \left| \nu_p^x - \nu_x^0 \right| \frac{g_p(\eta)/\lambda_p(\theta)}{\gamma_x/\gamma_n + g_p(\eta)/\lambda_p(\theta)} \right]^2 \quad (4.9)$$

Figure 4.5 illustrates the variation of  $\mathcal{J}(u_x)$  with respect to some of the variables that appear in (4.9). The  $x$  axis represents a wide range of positive values, since we do not have any prior information about this parameter. The plot is done in logarithmic scale for both axes.

The shape of  $\mathcal{J}(u_x)$  itself, when all the other parameters are fixed, is hard to explain, due to the cumbersome dependency with respect to  $u_x$ , which also intervenes in the terms  $\lambda_p(\theta)$ .

The SNR sweep shows that a high SNR corresponds to a higher amount of information. Figure 4.5b shows that  $\mathcal{J}(u_x)$  does not vary significantly with the width of the filter, in a case where  $\nu_x^0$  is close to the null frequency (and implicitly lies in the filter passband).

Figure 4.5c gives some insight concerning the information available in the various PSD and TF relative positioning cases. This plot shows that there will be two regimes: for very narrow PSDs the most advantageous situation information-wise is the case of total overlap (which implies a high  $\eta$ ). In the second regime (roughly when  $u_x > 10^{-3}$ )  $\mathcal{J}(u_x)$  is significantly higher in the case of a partial overlap. As for the other two situations, when  $u_x < \eta$  the information is higher in the total overlap case. Last, but not least, when  $u_x > \eta$ , the case of total overlap becomes the more challenging information-wise.

\*\*\*

A quantitative analysis of the Fisher information for the texture and TF parameters is harder to make as compared to the signal and noise levels. This is due to the more complex dependency, with multiple levels of non-linearity and interconnections. Despite this difficulty, we have provided an analysis that will aid in anticipating and understanding the method's performances.



### 4.2.2 Bayesian setting: priors, posterior and conditional posteriors

From the hierarchical conditioning of the variables, shown in Figure 4.1, the joint law for  $\mathbf{y}$ ,  $\mathbf{x}$  and  $\Psi$  writes:

$$\pi(\mathbf{y}, \mathbf{x}, \Psi) = f(\mathbf{y}|\mathbf{x}, \eta, \gamma_n) \cdot f(\mathbf{x}|\boldsymbol{\theta}, \gamma_x) \cdot \pi(\Psi) \quad (4.10)$$

where  $\pi(\Psi)$  represents the joint prior for the unknowns.

Using Bayes rule and taking into account that  $f(\mathbf{y})$  is constant with respect to the unknowns, the posterior is proportional to the joint law (4.10):

$$\begin{aligned} \pi(\mathbf{x}, \Psi|\mathbf{y}) \propto & \gamma_n^P \exp -\gamma_n \sum_{p=1}^P \left| \dot{y}_p - \dot{x}_p \dot{h}_p(\eta) \right|^2 \\ & \cdot \gamma_x^P \prod_{p=1}^P \lambda_p(\boldsymbol{\theta}) \cdot \exp \left[ -\gamma_x \sum_{p=1}^P |\dot{x}_p|^2 \lambda_p(\boldsymbol{\theta}) \cdot \pi(\Psi) \right] \end{aligned} \quad (4.11)$$

Due to the small amount of available information, there is no indication of a dependency between the priors of the variables, thus we can consider that the priors are independent of each other:

$$\pi(\Psi) = \pi(\boldsymbol{\theta}) \cdot \pi(\eta) \cdot \pi(\gamma_x) \cdot \pi(\gamma_n) \quad (4.12)$$

The forms for the priors can be judiciously chosen by analyzing (4.11). For  $\gamma_n$  the uninformative Jeffreys prior will be used, as explained in Section 2.2. It can be noticed that  $\gamma_x$  intervenes as a precision parameter in a Gaussian law and that in this case the Gamma law is the conjugate form:

$$\pi(\gamma_x|\alpha_x, \beta_x) \propto \gamma_x^{\alpha_x-1} \exp(-\beta_x \gamma_x) \quad (4.13)$$

Since there is no prior information concerning the values of the signal level, we should use an uninformative prior. Similarly as for  $\gamma_n$  a Jeffreys prior is obtained by setting  $(\alpha, \beta) \rightarrow (0, 0)$ .

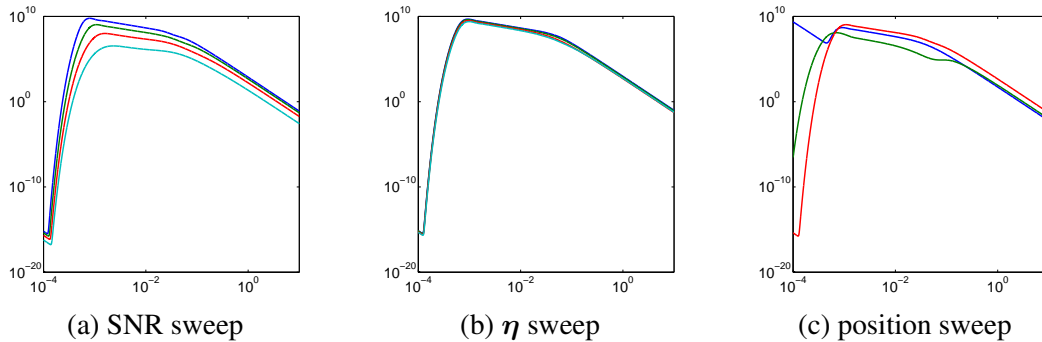


Figure 4.5: Fisher information for the PSD width  $u_x$  as a function of the SNR, the filter width and the relative positioning of the TF and the PSD. The legends are the same as those from Figure 4.3.



For  $\theta$  and  $\eta$ , the complicated dependency means that there is no conjugate form that can be used. Moreover, since there is no prior information regarding the values of these parameters, uninformative priors should also be used for the texture and instrument parameters. Consequently, uniform priors, defined over all the possible range of values, will be used for each parameter:

$$\begin{aligned}\pi(\theta) &= \mathcal{U}_{[\theta_m, \theta_M]}(\theta) \\ \pi(\eta) &= \mathcal{U}_{[\eta_m, \eta_M]}(\eta)\end{aligned}\tag{4.14}$$

Using the previously mentioned priors results in the following forms for the conditionals *a posteriori*:

$$\begin{aligned}\mathbf{x} &\sim \exp \left[ - \sum_{p=1}^P \left( \gamma_n |\dot{y}_p - \dot{h}_p(\boldsymbol{\eta}) \dot{x}_p|^2 + \gamma_x |\dot{x}_p|^2 \lambda_p(\boldsymbol{\theta}) \right) \right] \\ f(\mathbf{x}|\mathbf{y}, \gamma_n, \boldsymbol{\theta}) &- \text{separable and quadratic in the Fourier domain;} \\ \gamma_n &\sim \gamma_n^{P+\alpha_n-1} \cdot \exp \left[ -\gamma_n \left( \beta_n + \sum_{p=1}^P |\dot{y}_p - \dot{h}_p(\boldsymbol{\eta}) \dot{x}_p|^2 \right) \right] \\ f(\gamma_n|\mathbf{y}, \mathbf{x}, \alpha_n, \beta_n) &- \text{Gamma form;} \\ \gamma_x &\sim \gamma_x^{P+\alpha_x-1} \cdot \exp \left[ -\gamma_x \left( \beta_x + \sum_{p=1}^P |\dot{x}_p|^2 \lambda_p(\boldsymbol{\theta}) \right) \right] \\ f(\gamma_x|\mathbf{x}, \boldsymbol{\theta}, \alpha_x, \beta_x) &- \text{Gamma form;} \\ \boldsymbol{\theta} &\sim \prod_{p=1}^P \left\{ \lambda_p(\boldsymbol{\theta}) \cdot \exp \left[ -\gamma_x |\dot{x}_p|^2 \lambda_p(\boldsymbol{\theta}) \right] \right\} \cdot \mathcal{U}(\boldsymbol{\theta}) \\ f(\boldsymbol{\theta}|\mathbf{x}, \gamma_x) &- \text{independent on the observations, but has a very complicated dependency;} \\ \boldsymbol{\eta} &\sim \prod_{p=1}^P \left\{ \exp \left[ -\gamma_n |\dot{y}_p - \dot{h}_p(\boldsymbol{\eta}) \dot{x}_p|^2 \right] \right\} \cdot \mathcal{U}(\boldsymbol{\eta}) \\ f(\boldsymbol{\eta}|\mathbf{y}, \mathbf{x}, \gamma_n) &- \text{very complicated dependency.}\end{aligned}$$

**Remark:** The conditional law of the image with respect to the rest of the unknowns is closely related to Wiener filtering. In fact, the maximum of this law is the signal obtained through Wiener filtering. In the special case with no noise  $\gamma_n \rightarrow \infty$  the maximum is obtained by applying the inverse filter.

In order to determine the estimates the PM estimator is used, which is MSE optimal, as detailed in Appendix B. The estimates are obtained by calculating the integral  $\int_{\psi|\mathbf{y}} \psi \pi(\psi|\mathbf{y}) d\psi$ . The parameter dependency is complicated, making the integral intractable. In order to overcome this difficulty, and give the complicated laws for  $\boldsymbol{\theta}$  and  $\boldsymbol{\eta}$ , the Metropolis within Gibbs sampling technique will be employed to approximate the integral. Among the multiple versions for the MH step, the RWMH is chosen due to its simplicity and ease of implementation.

### 4.3 Results

As stated in the previous section, the iterative algorithm consists in sampling the parameters  $\Psi$  and  $\mathbf{x}$  and averaging the samples to obtain the PM estimates. The algorithm associated to our method is given in Algorithm 1. We consider this algorithm has converged once all the recursive means variations from one iteration to the next are less than 0.1% of the nominal parameter value.

---

**Algorithm 1:** Unsupervised Myopic Deconvolution for Textured Images Algorithm

---

```

input : Data  $\mathbf{y}$ 
output: Estimates for the texture parameters  $\theta$ , instrument parameters  $\eta$ ,
        hyperparameters  $\gamma_n$  and  $\gamma_x$ . In addition, samples for the texture  $\mathbf{x}^{(t)}$ 

% generate samples of  $\pi(\theta, \eta, \gamma_n, \gamma_x, \mathbf{x}|\mathbf{y})$  ;
 $t = 1$ ;
initialization  $\theta^{(t)}, \eta^{(t)}, \mathbf{x}^{(t)} = \mathbf{y}$ ;
while  $|\text{variation}| > \varepsilon$  do
     $t = t + 1$ ;

     $\gamma_n^{(t)} \sim f(\gamma_n|\mathbf{y}, \mathbf{x}^{(t-1)}, h_p(\eta^{(t-1)}), \alpha_n, \beta_n)$ 
     $\gamma_x^{(t)} \sim f(\gamma_x|\mathbf{x}^{(t-1)}, \theta^{(t-1)}, \alpha_x, \beta_x)$ 
     $\theta^{(t)} - \text{RWMH with target } f(\theta|\mathbf{x}, \gamma_x)$ 
     $\eta^{(t)} - \text{RWMH with target } f(\eta|\mathbf{y}, \mathbf{x}, \gamma_n)$ 
     $\mathbf{x}^{(t)} \sim f(\mathbf{x}|\mathbf{y}, \theta^{(t)}, \eta^{(t)}, \gamma_x^{(t)}, \gamma_n^{(t)})$ 
end

% Compute the parameter estimates by PM;
% let BI be the iteration where we consider the burn in period is over;

 $\hat{\theta} = \text{PM}(\theta^{(BI \dots T)})$ 
 $\hat{\eta} = \text{PM}(\eta^{(BI \dots T)})$ 
 $\hat{\gamma}_n = \text{PM}(\gamma_n^{(BI \dots T)})$ 
 $\hat{\gamma}_x = \text{PM}(\gamma_x^{(BI \dots T)})$ 

```

---

Table 4.1 lists the results of our estimation method, expressed in percentages (%). These results are a mean relative error on 20 realizations of each scenario for different parameter values. The  $\text{SNR} = 10 \log \gamma_n / \gamma_x$  represents the original signal to noise ratio. However, the Blurred SNR (used in [BMK09] to quantify the problem difficulty) is significantly smaller and depends on the positioning of the TF and PSD.

The data is more informative for some parameters, as shown by our Fisher information analysis in Appendix A and Section 4.2.1. This explains the difference of estimation er-

SNR	TF	$\gamma_n$	$\gamma_x$	$\nu_x^0$	$\nu_y^0$	$u_x$	$u_y$	$\eta$
20 dB	Narrow	1.2	3.2	4	4.2	8.5	8	4.3
	Wide	2.5	1.9	3	3.1	6.3	6.5	12
	Overlap	1.5	1.1	1.2	1.3	3.4	3.5	3
15 dB	Overlap	3.2	10.8	12.5	11.8	15.9	17.3	14.5
25 dB		1.9	1.6	1.1	0.9	3.1	3	2.7

Table 4.1: Parameter estimation MSE. The first part of the table shows the performances at a fixed SNR value and for different relative positioning of the PSD and TF. The second part of the table shows the method's sensitivity to noise by quantifying the error in favorable situations information-wise, for different SNR levels.

rors that can be observed in Table 4.1. The overall estimation performance for the central frequencies  $\{\nu_x^0, \nu_y^0\}$  are superior to those for the widths  $\{u_x, u_y\}$  and  $\eta$ . This is in direct correlation with the Fisher information shown in Figures 4.4 and 4.5. The Fisher information for the central frequencies has a relatively high value over the entire range of possible values, while in the case of the widths, it has a very high value for a range of values and it drops dramatically for the rest.

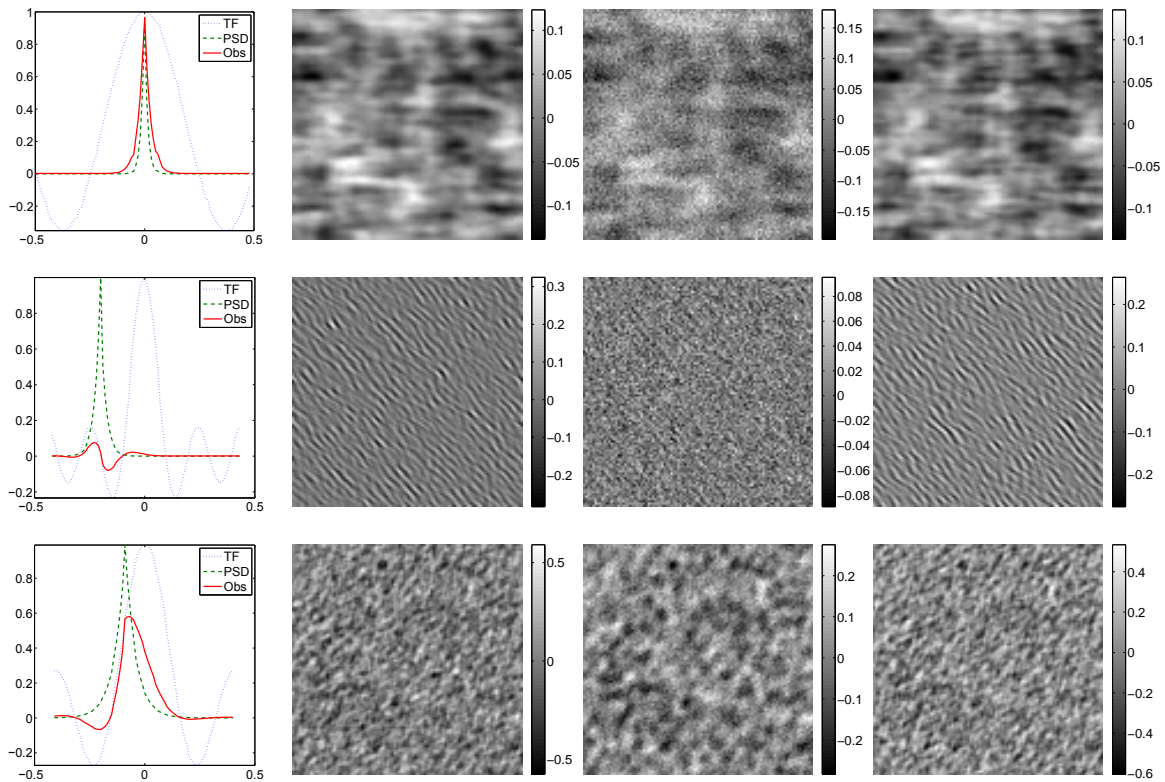


Figure 4.6: Illustrations of the algorithm performances for the cases presented in Figure 4.2. On the first row, the case with narrow, centered PSD and wide TF, on the second row, a PSD far from the null frequency and very narrow TF and on the third row, partial overlap.

The noise level is also an very important parameter, the accuracy of its estimation having a direct impact on the rest of the parameters. The higher noise levels are more accurately estimated, but the estimation of the rest of the parameters is very difficult in low SNR cases. From the signal level point of view, a high value means high SNR and triggers an accurate estimation for all the parameters, except for the noise level. Overall, a higher sensitivity to the noise than to the convolution can be noticed and this is clearly indicated by the estimation errors in Table 4.1.

We must stress the fact that despite the difficult cases, globally, the method yields a small error for all the parameters, proving the fact that it is well adapted for this problem and that it is able to provide accurate estimates even when the amount of available information is very reduced.

Figure 4.6 presents visual results for our algorithm. More specifically, the cases presented in Figure 4.2 are illustrated by showing on the first column the characteristics in the Fourier domain, on the second column, a texture corresponding to each case,  $x|\theta^*$ , on the third column, the degraded observations,  $y$ , and finally, on the last column, the reconstructed image  $x|\hat{\theta}$ , from which the blur and noise have been eliminated. The SNR = 25dB for all the cases, at this value the amount of information being sufficient to provide both a very good visual quality of the reconstruction and the restoration of the gray level scale. This shows the method's capacity to restore a textured image affected by a significant visual degradation, by correctly estimating the blur parameters and the noise level.

Figure 4.7 illustrates the method sensitivity to noise, by presenting the deconvolution results for the same texture, blurred by the same TF, but with different noise levels. On the first column we show the characteristics in the Fourier domain and the original texture  $x|\theta^*$ . On the following three columns we present successively the observations  $y$  (on top) and the deconvolution results  $x|\hat{\theta}$  (below). These cases correspond, from left to right to levels of SNR of 15dB, 20dB and 25dB.

From the information point of view, this is the most advantageous case, meaning that we should expect satisfactory results. Indeed, it can be noticed that the deconvolved images all resemble the original texture in what concerns the orientation and the frequency, meaning that the central frequencies of the PSD have been accurately estimated. However, for the cases with SNR = 15dB and SNR = 20dB, the textural content is very similar, but richer than the original texture, which indicates that the widths of the PSD have been slightly overestimated, resulting in a wider PSD. The reason for this phenomenon is that the algorithm attempts to adjust the thickness of the PSD tails, to account partly for the noise, in cases with high noise corruption.

Another aspect is related to the gray range, *i.e.*, to the estimation of  $\gamma_x$ . In the first two cases, the algorithm has a tendency to slightly overestimate the dynamic range of the observation, which means the  $\gamma_x$  is not as accurately estimated in these cases. However, in the SNR 25dB case,  $\gamma_x$  is correctly estimated.

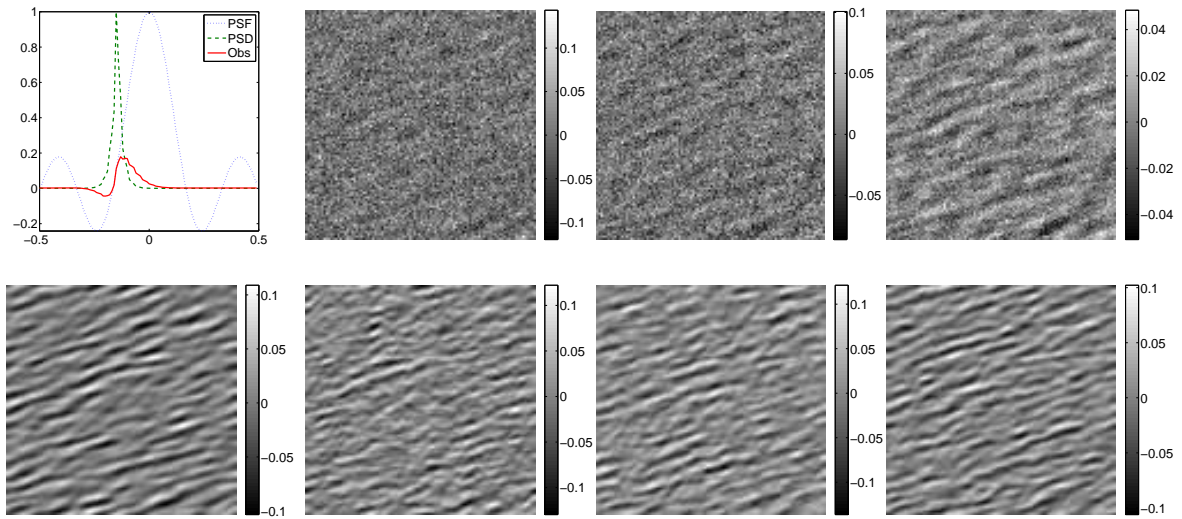


Figure 4.7: Example of observation system in the Fourier domain, with the corresponding observations and the deconvolution result, for various SNR values in a partial overlap case.

## 4.4 Conclusion and Perspectives

The main contribution presented in this chapter is the use of parametric models for the image and for the PSF, in order to tackle the deconvolution problem in a context of textured images with non-linear data dependency with respect to the PSF unknowns. To the best of our knowledge, there is no other method to address this problem. Moreover, by using the optimal PM estimator guarantees that at least from the MSE viewpoint, no other method can provide better results.

We have proposed a detailed analysis of the Fisher information for each unknown, which is directly related to the difficulty of the estimation. This has allowed us to anticipate the performances and then confirm them with numerical results.

Further developments to this problem consist in using more complex models for the PSF, which would be able to describe other types of distortions as well. For instance, this could allow us to address the problem of motion blur.



# Model Choice for the Law and the PSD of a Textured Image

## Contents

<b>5.1</b>	<b>Model Choice - State of the art</b>	<b>53</b>
<b>5.2</b>	<b>Problem Statement</b>	<b>54</b>
5.2.1	Texture coefficients law and PSD models	54
5.2.2	Probabilistic model choice	56
5.2.3	Joint law and priors for the model, image and parameters	57
<b>5.3</b>	<b>Evidence calculation</b>	<b>58</b>
5.3.1	Evidence approximations based on posterior samples	58
5.3.2	Posterior sampling	60
5.3.3	Gibbs within-model posterior sampling	61
5.3.4	Implementation issues	63
<b>5.4</b>	<b>Experimental Results</b>	<b>63</b>
5.4.1	RWMH vs FMH	65
5.4.2	HMA and LMA	66
5.4.3	CEAPS performances	66
5.4.4	Visual reconstructions	68
<b>5.5</b>	<b>Conclusion and Perspectives</b>	<b>70</b>

Chapter 4 has presented a first subproblem in our textured images restoration context. This chapter is devoted to another interesting problem: texture model selection from indirect data. Our main contribution is tackling a problem that has not been dealt with so far by developing a method based on an optimal risk decision.

## 5.1 Model Choice - State of the art

Model choice applications span over a wide range of fields, for instance microbiology, proteomics, economics, statistics, signal and image analysis. The large majority of these methods are based on the log-likelihood value computed for the Maximum Likelihood Estimate (MLE),  $\hat{\theta}$  and contain different penalization terms for the model dimension.

The Generalized MLE (GMLE) method for model selection consists in determining the model with the highest likelihood for the MLEs of the parameters. However, this



estimator cannot be used in the case of incomplete data and thus it is not applicable for indirect observations or the non-Gaussian texture model (due to the presence of the auxiliary variables).

Among the most frequently employed classifiers is the well-known MLE-based Akaike Information Criterion (AIC) [Aka74], with penalization term  $2D_k$  where  $D_k$  is the dimension of model  $k$ . Equally popular is the Bayesian approach, with the Deviance Information Criterion [SBCL02] and the Bayesian Information Criterion (BIC) [Sch78], also based on the maximized value of the likelihood and a penalization term of the form  $D_k \ln N$ , where  $N$  is number of observations. In the same class of Bayesian approaches are included the Bayes factors and the method used in this work, based on the evidence. This quantity is mostly known in the literature as the marginal likelihood, since it is in fact obtained by marginalization. However, for brevity, we will systematically be referring to it as the evidence (in favor of a certain model).

An important aspect is model complexity penalization. While the AIC tends to select too complex models, the BIC is prone to underfitting. Hence, from a methodological perspective, being able to quantify the performances of the model choice algorithm is crucial. For this reason, using a method that is optimal in some sense offers the certainty that, at least from that point of view, it cannot be outperformed by any concurrent approach.

Moreover, all these criteria are based on approximations. The evidence based methods represent an alternative Bayesian approach [Bea03] that does not rely on an approximation and can be formulated under the form of Bayes factors or the posterior model probabilities computation method presented in this work. This method is based on the formulation of an optimal decision function from the Bayes risk viewpoint. The form of the Bayes risk and details concerning the computation of this classifier are given in Appendix B. The optimality is achieved due to the fact that the cost is averaged over all possible data realizations, true models and parameter values. Moreover, for a binary cost function  $C(k, k^*) = 1 - \delta(k, k^*)$ , the method implicitly selects the model with the MAP probability.

## 5.2 Problem Statement

Our problem consists in texture model selection from an indirectly observed realization. More precisely, textures modeled by GRFs or SMGRFs, with PSDs taking various parametric forms, are observed via an imperfect system that introduces a blur and noise. Starting from these observations, the goal is to select the model that best describes the texture.

### 5.2.1 Texture coefficients law and PSD models

The texture model, say  $\mathcal{M} = k^*$ , is chosen among  $K$  possible models. The same notations used in the previous chapters are employed here for the unobserved image  $\mathbf{x}$ , the observations  $\mathbf{y}$  and the noise precision  $\gamma_n$ . The texture parameters corresponding to model  $\mathcal{M} = k$  are denoted by  $\gamma_k$ ,  $\mathbf{s}_k$  and  $\boldsymbol{\theta}_k$ .



$\mathcal{M}$	Model $k$	Expression of $\lambda^k(\nu_x, \nu_y, \theta_k)$	$\theta_k$
1	Lorentz	$\left[1 + \frac{(\nu_x - \nu_{x0})^2}{\sigma_x^2} + \frac{(\nu_y - \nu_{y0})^2}{\sigma_y^2}\right]$	$\nu_{x0}, \nu_{y0}, \sigma_x, \sigma_y$
2	Generalized Lorentz	$\left[1 + \frac{(\nu_x - \nu_{x0})^2}{\sigma_x^2} + \frac{(\nu_y - \nu_{y0})^2}{\sigma_y^2}\right]^q$	$\nu_{x0}, \nu_{y0}, \sigma_x, \sigma_y, q$
3	Exponential	$\exp \frac{1}{2} \left[ \frac{ \nu_x - \nu_{x0} }{\sigma_x} + \frac{ \nu_y - \nu_{y0} }{\sigma_y} \right]$	$\nu_{x0}, \nu_{y0}, \sigma_x, \sigma_y$
4	Generalized Gauss	$\exp \frac{1}{2} \left[ \frac{ \nu_x - \nu_{x0} ^q}{\sigma_x^q} + \frac{ \nu_y - \nu_{y0} ^q}{\sigma_y^q} \right]$	$\nu_{x0}, \nu_{y0}, \sigma_x, \sigma_y, q$

Table 5.1: PSD models, the expression of the  $\lambda$  coefficients and the corresponding parameters.

One of the most important features of any efficient model selection method is the ability to penalize the model complexity, more specifically to be capable of selecting the less complex model for similar measures of model fit. In order to test the method's performances from the complexity penalization point of view we have chosen to include embedded models in our dictionary of shapes for the PSD.

The texture PSD models are built in the reduced frequency domain. There is virtually no constraint on the form, except for positivity: it can be constant, corresponding to a white noise, or even a single frequency field. Here, the focus is on parametric, unimodal functions, with a relatively reduced number of parameters.

The chosen parametric shapes are: Lorentzian ( $\mathcal{M} = 1$ ), Generalized Lorentzian ( $\mathcal{M} = 2$ ), Exponential ( $\mathcal{M} = 3$ ), and Generalized Gaussian ( $\mathcal{M} = 4$ ). Each model  $k$  is driven by the corresponding  $\theta_k$ . Table 5.1 shows explicitly the dependencies  $\lambda^k(\theta_k)$ . The parameters of these parametric forms are: the central frequencies  $\nu_{x0}, \nu_{y0}$  and the widths  $\sigma_x, \sigma_y$ .  $q$  represents the power parameter and is specific to models  $\mathcal{M} = 2$  and  $\mathcal{M} = 4$ .

We have chosen to use embedded models, for instance  $\mathcal{M} = 1$  and  $\mathcal{M} = 3$  are nested in  $\mathcal{M} = 2$  and  $\mathcal{M} = 4$ , respectively. This enables an analysis on the method's capacity to penalize model complexity when the extra parameters do not trigger a significant model fit increase.

The textured images are spatially discrete, thus the PSD is defined on the reduced frequency domain, *i.e.*, the variables  $(\nu_x, \nu_y) \in [-0.5, 0.5]^2$ . Furthermore, let us consider that the Fourier coefficient  $p$  has the  $(\nu_m, \nu_n)$  position in the discrete reduced frequency domain. The  $\lambda_p$  from (5.4) are the elements of the PSD field at these discrete positions and depend on the model,  $\mathcal{M} = k$ , and on the parameters  $\theta_k$ . From this point forward, to explicitly show this dependency, the notation  $\lambda_p^k(\theta_k)$  is used and, more precisely:  $\lambda_p^k(\theta_k) = \lambda^k(\nu_m, \nu_n, \theta_k)$ .

The use of parametric models has the advantage of reducing the number of unknowns. On the other hand, this model defines a highly non-linear dependency of  $\lambda_p^k(\theta_k)$  with respect to  $\theta_k$ , as shown in Table 5.1. This complicated dependency means that there is no

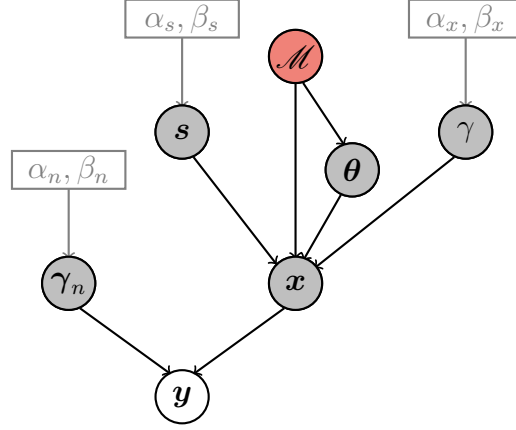


Figure 5.1: Hierarchical variable dependency for the texture model choice from indirect observations.

conjugate form for this law.

Moreover, the prior information about the parameters is very reduced, thus, uninformative priors will be used. Consequently, a uniform prior is employed:  $\pi(\theta_k | \mathcal{M} = k) = \mathcal{U}_{[\theta_k^m, \theta_k^M]}(\theta_k)$ .

### 5.2.2 Probabilistic model choice

This model choice problem is twofold and refers to:

1. the form of the law for the texture coefficients, *i.e.*, whether we are dealing with a GRF or a SMGRF,
2. the parametric form of the PSD.

The difficulty of this problem is mainly due to the fact that the image models (presented in Chapter 3) are highly non-linear and have a very complicated dependency with respect to the parameters. Moreover, the parameters driving these models are unknown. Consequently, in order to infer on the models, we also have to infer on their values. Therefore, this is far from trivial even in the case of direct observations. Furthermore, the fact that we are dealing with blurred and noisy observations adds a supplementary layer of complexity.

The indirect observations aspect is not circumvented by the fact that the PSF is known, since the noise level  $\gamma_n$  and the signal level  $\gamma$  are unknown. This is all the more challenging especially in our textured image context, where the blur and noise corruption yield a new textured image. This new image can be mistakenly considered either as being a realization of the same texture model as the original image, but with different parameter values  $\theta$ , to account for the blur and noise, or even of a different PSD model, with thicker tails, to account for high levels of noise. Consequently, the additional difficulty consists in not being misled by the distortion and correctly distinguishing the true model  $\mathcal{M}^*$ , the true parameter values  $\theta^*$  and the true hyperparameter values  $\gamma_n^*$  and  $\gamma^*$ .

Figure 5.1 shows the variable conditioning for this problem. The red singles out the texture model, which is in fact the quantity of interest. In gray, we have represented the rest

of the unknowns, whose estimation is not our primary goal. However, our model choice algorithm will provide samples for these unknowns (conditionally on the model), as an additional, auxiliary result.

The detailed presentation of this problem and the method we propose for solving it has made the subject of our paper [VGB].

This method relies on a probabilistic framework. Using Bayes' rule, the posterior model probabilities are:

$$\Pr(\mathcal{M} = k | \mathbf{y}) = \frac{f(\mathbf{y} | \mathcal{M} = k) \cdot p_k}{f(\mathbf{y})} \quad (5.1)$$

and require the computation of two quantities.

- i. The probability distribution of the data,  $f(\mathbf{y})$ . Fortunately, it does not depend on the model, thus can be calculated by normalization.
- ii. The evidence,  $e_k = f(\mathbf{y} | \mathcal{M} = k)$ , obtained from the joint law of data and unknowns, given the model, by marginalizing the unobserved texture, the noise precision and the texture parameters:

$$e_k = \int_{\Psi} f(\mathbf{y}, \Psi | \mathcal{M} = k) d\Psi \quad (5.2)$$

where  $\Psi$  represents all the unknowns.

The optimal classifier is built in a Bayesian framework, where each model's posterior probability is determined from the model evidences. These evidences are intractable and thus are numerically computed by MCMC methods.

From an algorithmic viewpoint, this work embeds the FMH algorithm introduced in [VGB11] and presented in Section 2.4.5. Consequently, an important performance increase is achieved, due to the specific nature of our problem. In our case, there is no need for computing second order derivatives and thus a Newton type proposal is built using only first order derivatives.

### 5.2.3 Joint law and priors for the model, image and parameters

Our model selection method relies on a probabilistic formulation of the problem in a Bayesian framework and determines the *a posteriori* probability of each model  $\Pr(\mathcal{M} = k | \mathbf{y})$ . The *a priori* distribution for the model is fully described by the  $p_k = \Pr(\mathcal{M} = k)$  probabilities. In our numerical study, we have employed an uninformative prior, *i.e.*, *a priori* equiprobable models:  $p_k = 1/K$ ,  $k = 1 \dots K$ .

The parametric texture models are driven by the parameter set  $\chi$ . Thus, for each model,  $k$ , we have the corresponding parameter set  $\chi_k$  and the law  $f(\mathbf{x} | \chi_k, \mathcal{M} = k)$ .

Let  $\Psi = \{\gamma_n, \mathbf{x}, \chi_k\}$  represent all the unknowns. The joint law is written using the

conditioning rule and the hierarchy shown in Figure 5.1:

$$f(\mathbf{y}, \Psi | \mathcal{M} = k) = f(\mathbf{y} | \mathbf{x}, \gamma_n) \cdot f(\mathbf{x} | \chi_k, \mathcal{M} = k) \cdot \pi(\chi_k | \mathcal{M} = k) \cdot \pi(\gamma_n) \quad (5.3)$$

The  $f(\mathbf{x} | \chi_k, \mathcal{M} = k)$  represents the law for the texture, driven by the parameters  $\chi_k = \{\gamma_k, \mathbf{s}_k, \boldsymbol{\theta}_k\}$  and the model index  $\mathcal{M}$ . Then, the expression of this law is:

$$f(\mathbf{x} | \gamma_k, \mathbf{s}_k, \boldsymbol{\theta}_k, \mathcal{M} = k) = \gamma_x^P \prod_{p=1}^P [s_{kp} \lambda_p^k(\boldsymbol{\theta}_k)] \cdot \exp - \sum_{p=1}^P [\gamma_x s_{kp} \lambda_p^k(\boldsymbol{\theta}_k) | \hat{x}_p|^2] \quad (5.4)$$

The priors  $\pi(\gamma_k)$ ,  $\pi(\mathbf{s}_k)$  and  $\pi(\boldsymbol{\theta}_k | \mathcal{M} = k)$  are explicitly given in Section 5.2.1, while  $\pi(\gamma_n)$  is specified in Section 2.2.

### 5.3 Evidence calculation

The full description of our Bayesian model choice method relies (i) on the data model and (ii) on the priors for the unknowns. The hierarchical direct model and the texture model specificities are shown in Figure 5.1. In the following, the emphasis is on the SMGRF, which encompasses the GRF. By writing (5.2) as:

$$e_k = \int_{\mathbf{x}, \gamma_k, \gamma_n, \boldsymbol{\theta}_k, \mathbf{s}_k} f(\mathbf{y} | \Psi, \mathcal{M} = k) \cdot \pi(\Psi | \mathcal{M} = k) d\mathbf{s}_k d\boldsymbol{\theta}_k d\gamma_k d\gamma_n d\mathbf{x} \quad (5.5)$$

and plugging in (5.11) the intractability of the integral is obvious. For this reason, it must be calculated numerically and the solution chosen here is sampling.

A natural idea is to straightforwardly determine the evidence  $e_k$  from samples of the prior  $\pi(\Psi | \mathcal{M} = k)$ :  $\Psi^{(t)} = (\mathbf{x}^{(t)}, \gamma_n^{(t)}, \gamma_k^{(t)}, \mathbf{s}_k^{(t)}, \boldsymbol{\theta}_k^{(t)})$ ,  $t = 1 \dots T$  as follows:

$$\bar{e}_k = \frac{1}{T} \sum_{t=1}^T f(\mathbf{y} | \Psi^{(t)}, \mathcal{M} = k) \quad (5.6)$$

It consists in sampling the priors and computing the arithmetic mean of the corresponding likelihood values – Arithmetic Mean Approximation (AMA). Evidence computation based on prior samples can also be done by nested sampling [Ski06].

Nevertheless, when the likelihood is very peaked, as in the current case, most of these samples will have weak likelihood, *i.e.*, an insignificant contribution and thus, the algorithm is slow to converge. For this reason, it is more suitable to compute (5.5) by sampling the posterior  $\pi(\Psi, \mathcal{M} = k | \mathbf{y})$ .

#### 5.3.1 Evidence approximations based on posterior samples

Our model selection method is based on evidence approximation from posterior samples and the proposed method will be referred to as the Classifier based on Evidence

Approximation from Posterior Samples (CEAPS). This method can be formulated using two different approximations for the evidence, presented in the following: the Harmonic Mean Approximation (HMA) [NR94] and the Laplace-Metropolis Approximation (LMA) [Raf95].

### 5.3.1.1 Harmonic Mean Approximation

Let us consider that  $\Psi^{(t)} = (\mathbf{x}^{(t)}, \gamma_n^{(t)}, \gamma_k^{(t)}, \mathbf{s}_k^{(t)}, \boldsymbol{\theta}_k^{(t)})$  with  $t = 1 \dots T$  are samples from the *a posteriori* law. Then, the evidence can be computed as:

$$\tilde{e}_k \approx \left\{ \frac{1}{T} \sum_{t=1}^T \left[ f(\mathbf{y} | \Psi^{(t)}, \mathcal{M} = k) \right]^{-1} \right\}^{-1} \quad (5.7)$$

i.e., the harmonic mean of the likelihood values for the samples  $\Psi^{(t)}$ .

Although  $\tilde{e}_k$  converges almost surely to the true value  $e_k$  when  $T \rightarrow \infty$  [Pak99], it does not generally satisfy the central limit theorem [RW09]. Occasionally, a  $\Psi^{(t)}$  with significant *a priori* probability, but very low likelihood, may occur. Its contribution in the harmonic mean is high and this may trigger infinite variances [NR94]. Solutions to stabilize this estimator have been provided in [RNSK07]. Nevertheless, we have not encountered this difficulty in neither of our model choice works [VGR12, VGB], where the priors are uninformative on a finite interval and the likelihood is very peaked. In these cases, the posterior samples are distributed in the regions where the likelihood has significant values. Consequently, the situations where the HMA may diverge or converge too slowly are avoided.

Moreover, its computational efficiency and ease of implementation have lead to its use in a series of topic modeling papers such as [GS04, Wal06].

### 5.3.1.2 Laplace-Metropolis Approximation

The evidence (5.5) can also be expressed as:

$$e_k = \int_{\Psi} \exp \left\{ \underbrace{\log [f(\mathbf{y} | \Psi, \mathcal{M} = k) \cdot \pi(\Psi | \mathcal{M} = k)]}_{F_k(\Psi, \mathbf{y})} \right\} d\Psi \quad (5.8)$$

with  $F_k(\Psi, \mathbf{y})$  the log-posterior computed for observation  $\mathbf{y}$ .

**Remark:**  $F_k(\Psi, \mathbf{y})$  is related to the Fisher information  $\mathcal{J}_k$  and indicates the mean amount of available information, for the given observation  $\mathbf{y}$ .  $F_k(\Psi, \mathbf{y})$  is related to the Fisher information  $\mathcal{J}_k$  introduced in Chapter 2 and Appendix A through (A.1) as follows:

$$\mathcal{J}_k(\Psi) = -\mathbb{E}_{\mathbf{y}} [F_k''(\Psi, \mathbf{y})] \quad (5.9)$$

Under the hypothesis that  $F_k(\Psi, \mathbf{y})$  is twice differentiable with a unique maximum in  $\Psi^*$ , a Laplace approximation can be applied to evaluate the integral (5.8):

$$\hat{e}_k \approx \exp [PF_k(\Psi_k^*, \mathbf{y})] \cdot (2\pi/P)^{D_k/2} \cdot \left| -F_k''(\Psi_k^*, \mathbf{y}) \right|^{-1/2} \quad (5.10)$$

where  $\Psi_k^*$  represents the MAP value for model  $k$ ,  $F_k''(\Psi_k^*, \mathbf{y})$  is the Hessian of the log-posterior, evaluated at  $\Psi_k^*$  and  $D_k$  is the dimension of model  $k$ . The last factor in (5.10) is the determinant of the observed information matrix. In fact, computing the evidence in this manner consists in determining the MAP value,  $\Psi^*$ , *i.e.*, the value for which  $F(\Psi, \mathbf{y})$  is maximum, and replacing this value in relation (5.10). [KR95] reviews the Laplace based methods for evidence computation. These approximation methods have relative errors of order  $\mathcal{O}(P^{-1})$ .

This approximation can also be performed based on MCMC samples from the posterior. In this case the method is called Laplace Metropolis Approximation (LMA) [Raf95]. The LMA can be based in the MAP, the PM, or the MedAP and the Hessian can be approximated by the covariance matrix of the samples. In our case, this approximation is performed using the PM and the value of the Hessian computed for the PM.

**Remark:** The LMA explicitly penalizes complex models due to the second factor that decreases exponentially with model dimension, while for the HMA the model complexity penalization is implicit, achieved through the likelihood values used to compute the evidence.

### 5.3.2 Posterior sampling

The posterior law is proportional to the joint law:

$$\begin{aligned} f(\mathbf{y}, \Psi, \mathcal{M} = k) = & \mathcal{C} \cdot \exp \left[ -\gamma_n \sum_{p=1}^P |\overset{\circ}{y}_p - \overset{\circ}{h}_p \overset{\circ}{x}_p|^2 \right] \\ & \cdot \gamma_n^{P+\alpha_n-1} \exp [-\beta_n \gamma_n] \cdot \prod_{p=1}^P [s_{kp} \cdot \lambda_p^k(\theta_k)] \\ & \cdot \gamma_k^{P+\alpha_x-1} \exp [-\beta_x \gamma_k] \exp \left[ -\gamma_k \sum_{p=1}^P |\overset{\circ}{x}_p|^2 s_{kp} \lambda_p^k(\theta_k) \right] \\ & \cdot \mathcal{U}_{[\theta_k^m, \theta_k^M]}(\theta_k) \cdot \prod_{p=1}^P s_{kp}^{\alpha_s-1} \cdot \exp \left[ -\beta_s \sum_{p=1}^P s_{kp} \right] \end{aligned} \quad (5.11)$$

where the normalization constant is  $\mathcal{C} = K^{-1} \cdot (2\pi)^{-3P} \cdot \beta_n^{\alpha_n} \cdot \Gamma^{-1}(\alpha_n) \cdot \beta_x^{\alpha_x} \cdot \Gamma^{-1}(\alpha_x) \cdot \beta_s^{P\alpha_s} \cdot \Gamma(\alpha_s)^{-P} \cdot (\theta_k^M - \theta_k^m)^{-1}$ .

However, this law cannot be directly sampled, thus MCMC methods will be employed, more precisely, Gibbs sampling. This can be performed via two types of algorithms.

- Across-model approach – joint sampling of the model index and its parameters. The algorithm jumps from one model to another and explores the joint model index plus parameter space, yielding a joint chain of model indexes and parameter values. The most representative algorithm of this type is Reversible Jump MCMC (RJMCMC) [Gre95].
- Within-model approach – consists in exhaustively visiting the candidate models and parameter sampling conditionally on the model. It provides  $K$  chains of parameter values, one for each model. For a detailed description see [GD94, NR94] and the more recent survey [RW09].

Despite the conceptual differences, for a finite candidate models set, the two approaches yield the same result (provided they have reached convergence) but, under two different forms.

The RJMCMC algorithm is especially interesting for very large numbers of models, when an exhaustive sequential sampling of all the models may be prohibitively expensive. Nevertheless, this algorithm may pose problems when the models are very different. In this case, when skipping from one model to another, a transformation must be applied in order to determine the current values of the parameters for the new model, based on the current parameter values for the old model. This transformation may be difficult, or even impossible to determine analytically. Furthermore, applying an incorrect transformation may trigger high rejection rates and may lead to an inefficient exploration of the model-parameter space.

Since in our problem the number of concurrent models is rather reduced, the within-model sampling is the best strategy. This avoids as well the non-trivial RJMCMC problems concerning the parameter transformation when switching models. Moreover, the within-model approach guarantees that all models have been thoroughly explored and the model selection is not affected by the sampling algorithm. For this reason, our model choice method is based on within-model posterior sampling.

### 5.3.3 Gibbs within-model posterior sampling

The samples from the posterior law are obtained using Gibbs sampling. Among the various strategies, we have chosen to sample  $\gamma_n, \gamma_k, \mathbf{x}, \boldsymbol{\theta}_k$  and  $\mathbf{s}_k$ . The advantage of this approach is that we obtain rather standard targets and we can perform parallel sampling for  $\mathbf{x}$  and  $\mathbf{s}$ . Then, the *a posteriori* conditional laws for the parameters are:

$$\mathbf{x} \sim \prod_p \exp \left[ - \left( \gamma_n |\hat{y}_p - \hat{h}_p \hat{x}_p|^2 + \gamma_k |\hat{x}_p|^2 s_{kp} \lambda_p^k(\boldsymbol{\theta}_k) \right) \right] = \prod_p \mathcal{N}(m_p, v_p)$$

$$\text{with } m_p = \gamma_n \hat{y}_p \hat{h}_p v_p$$

$$v_p = \left( \gamma_n g_p + \gamma_k s_{kp} \lambda_p^k(\boldsymbol{\theta}_k) \right)^{-1}$$

where  $g_p = |\hat{h}_p|^2$ .

- separable in the Fourier domain, *i.e.*, parallel sampling is possible,
- computation cost equivalent to sampling the *a priori* law;

$$\mathbf{s}_k \sim \prod_p s_{kp}^{\alpha_s} \cdot \exp \left\{ - \sum_{p=1}^P s_{kp} [\gamma_x |\dot{x}_p|^2 \lambda_p^k(\boldsymbol{\theta}_k) + \beta_s] \right\} = \prod_p \mathcal{G}(a_s, b_{sp})$$

$$\text{with } a_s = \alpha_s + 1$$

$$b_{sp} = \beta_s + \gamma_k |\dot{x}_p|^2 \lambda_p^k(\boldsymbol{\theta}_k)$$

– separable, independent on the observations, allowing for parallel sampling;

$$\gamma_n \sim \gamma_n^{P+\alpha_n-1} \cdot \exp \left[ -\gamma_n \left( \beta_n + \sum_p |\dot{y}_p - \dot{h}_p \dot{x}_p|^2 \right) \right] = \mathcal{G}(a_n, b_n)$$

$$\text{with } a_n = \alpha_n + P$$

$$b_n = \beta_n + \sum_p |\dot{y}_p - \dot{h}_p \dot{x}_p|^2$$

$$\gamma_k \sim \gamma_k^{P+\alpha_x-1} \cdot \exp \left[ -\gamma_k \left( \beta_x + \sum_p |\dot{x}_p|^2 s_{kp} \lambda_p^k(\boldsymbol{\theta}_k) \right) \right] = \mathcal{G}(a_x, b_x)$$

$$\text{with } a_x = \alpha_x + P$$

$$b_x = \beta_x + \sum_p |\dot{x}_p|^2 s_{kp} \lambda_p^k(\boldsymbol{\theta}_k)$$

$$\boldsymbol{\theta}_k \sim \prod_p \left\{ \lambda_p^k(\boldsymbol{\theta}_k) \cdot \exp \left[ -\gamma_k s_{kp} |\dot{x}_p|^2 \lambda_p^k(\boldsymbol{\theta}_k) \right] \right\} \cdot \mathcal{U}(\boldsymbol{\theta}_k)$$

– very complicated dependency.

**Remark:** The alternative to sampling all the unknowns is to integrate a part of them, in order to avoid certain sampling steps:

- $\mathbf{x}$  marginalization and sampling the rest of the unknowns, thus no texture sampling, but even more cumbersome dependency on  $\boldsymbol{\chi}_k$  and  $\gamma_n$ . This strategy is similar to the collapsed Gibbs sampler method used in [KTHD12].  $\mathbf{s}$  remains separable, thus parallel sampling is feasible, but not of standard Gamma laws. Furthermore, the posterior for  $\gamma_n$  and  $\gamma_k$  no longer have Gamma forms either:

$$f(s_{kp}, \gamma_n, \boldsymbol{\theta}_k | *) \propto \exp \left[ \frac{1}{r_p} \frac{|\dot{y}_p|^2 g_p}{\gamma_k s_{kp} \lambda_p(\boldsymbol{\theta}_k)} \right] \cdot \exp - \left[ r_p \gamma_n \gamma_k s_{kp} \lambda_p(\boldsymbol{\theta}_k) \right] \left[ \dot{x}_p - \frac{1}{r_p} \frac{\dot{y}_p^* \dot{h}_p}{\gamma_k s_{kp} \lambda_p(\boldsymbol{\theta}_k)} \right]^2$$

- $s_k$  marginalization, resulting in a complicated law for  $\mathbf{x}$  (loss of the upper hand of SMGRF texture model – the conditional Gaussianity for  $\dot{x}_p$ ). However,  $\dot{x}_p$  remain



independent, thus can be sampled in parallel, but by more sophisticated samplers. The new conditional laws are:

$$f(x_p|*) \propto \left[ 1 + \frac{\gamma_k \lambda_p^k(\theta_k)}{\beta_s} |\hat{x}_p|^2 \right]^{-\alpha_s-1} \exp \left[ -\gamma_n |\hat{y}_p - \hat{h}_p \hat{x}_p|^2 \right]$$

- integrate both  $\mathbf{x}$  and  $\mathbf{s}_k$  and sample only  $\gamma_n$ ,  $\gamma_k$  and  $\theta_k$ , which will be distributed under complicated laws.

Moreover, by variable marginalization, the resulting law is more diluted, thus what is gained by eliminating sampling steps, may be lost in terms speed of convergence. We have chosen not to integrate any of them since, although this implies more sampling steps, the sampled laws are easier to handle.

Since the law for  $\theta$  has a non-standard, complicated form, it can not be sampled directly and a Metropolis-within-Gibbs strategy, similar to the one in Chapter 4, is employed. However, in this case, the efficient FMH is used.

Hence, it is obvious that sampling the *a posteriori* conditional laws implies an extra computational effort, both in terms of deployed algorithms, and from the law complexity point of view. Nevertheless, the advantages are significant, the samples no longer having insignificant likelihood. This considerably reduces the number of samples needed to reliably compute the evidences.

### 5.3.4 Implementation issues

The algorithm implementation has raised a series of numerical problems. Since the likelihood has an exponential form for each Fourier coefficient and consists in a multiplication over all the coefficients, this quantity often exceeds Matlab's representation capabilities and is set to infinity. To overcome this obstacle, the Negative Log-Likelihood (NLL) is computed instead of the likelihood for evaluating the acceptance probabilities in the MH algorithm. However, the likelihood is still needed for the evidence computation (both by the HMA and the LMA) and the indetermination problems caused by the variables set to infinity or zero [VGR12] are again encountered. The employed solution is to determine the minimum value of each NLL chain, subtract it from all the NLL of the chain and compute the evidence as the harmonic mean or the LMA based on the "offset" values. The normalization is reversed in the final stage of posterior probability computation. The resulting CEAPS algorithm is given in Algorithm 2.

## 5.4 Experimental Results

The present section is devoted to describing and discussing the performances of our CEAPS model selection method for blurred and noisy textured images. A series of tests will be presented in the following:

---

**Algorithm 2:** Classifier based on Evidence Approximations from Posterior Samples (CEAPS) algorithm

---

**input** : Data  $\mathbf{y}$ , models dictionary for  $\mathcal{M} = k$ ,  $k = 1..K$   
**output:** Evidences  $\tilde{e}_k$  + samples for texture  $\mathbf{x}^{(t)}$ , noise parameter  $\gamma_n^{(t)}$ , texture parameters  $\gamma_k^{(t)}$ ,  $\mathbf{s}_k^{(t)}$ ,  $\boldsymbol{\theta}_k^{(t)}$  ( $t = 1..T$ )

*% prior model probabilities, e.g.,:*  
 $p_k = 1/K$ ;

*% generate samples of  $\pi(\boldsymbol{\theta}_k, \mathbf{x}, \mathbf{s}, \gamma_n | \mathbf{y}, \mathcal{M} = k)$  ;*

**for**  $k = 1$  **to**  $K$  **do**

*% Gibbs sampler for  $(\gamma_n, \gamma_k, \mathbf{s}_k, \boldsymbol{\theta}_k, \mathbf{x})$ ,  $\mathcal{M} = k$  fixed:*  
 $t = 1$ ;  
initialization  $\boldsymbol{\theta}_k^{(t)}, \mathbf{x}^{(t)} = \mathbf{y}, \mathbf{s}^{(t)}$ ;  
 $\text{NLL}_k(t) = \text{NegLogL}(\mathbf{x}^{(t)}, \mathbf{s}^{(t)}, \boldsymbol{\theta}_k^{(t)})$ ;  
 $m(k) = \min(\text{NLL}_k)$ ;  
 $\text{nNLL}_k = \text{NLL}_k - m(k)$ ;  
*% compute the evidence using HMA or LMA*  
 $\tilde{e}_k = \text{ComputeEvidence}(\text{nNLL}_k)$ ;  
 $e_{rec}(t) = 0$ ;

**while**  $|e_{rec}(t) - \tilde{e}_k| > \varepsilon$  **do**

$t = t + 1$ ;

$\gamma_n^{(t)} \sim f(\gamma_n | \mathbf{y}, \mathbf{x}^{(t-1)}, \alpha_n, \beta_n)$   
 $\gamma_k^{(t)} \sim f(\gamma_k | \mathbf{x}^{(t-1)}, \boldsymbol{\theta}_k^{(t-1)}, \mathbf{s}_k^{(t-1)}, \alpha_x, \beta_x)$   
 $\mathbf{s}_k^{(t)} \sim f(\mathbf{s}_k | \mathbf{x}^{(t-1)}, \boldsymbol{\theta}_k^{(t-1)}, \alpha_s, \beta_s)$   
 $\boldsymbol{\theta}_k^{(t)} - \text{FMH with target } f(\boldsymbol{\theta}_k | \mathbf{x}^{(t-1)}, \mathbf{s}_k^{(t)})$   
 $\mathbf{x}^{(t)} \sim f(\mathbf{x} | \mathbf{y}, \gamma_k^{(t)}, \gamma_n^{(t)}, \mathbf{s}_k^{(t)}, \boldsymbol{\theta}_k^{(t)})$

$\text{NLL}_k(t) = \text{NegLogL}(\mathbf{x}^{(t)}, \gamma_k^{(t)}, \mathbf{s}_k^{(t)}, \boldsymbol{\theta}_k^{(t)})$ ;  
 $m(k) = \min(\text{NLL}_k)$ ;  
 $\text{nNLL}_k = \text{NLL}_k - m(k)$ ;  
 $\tilde{e}_k = \text{ComputeEvidence}(\text{nNLL}_k)$ ;  
 $e_{rec}(t) = \text{RecursiveEvidence}(e_{rec}(t-1), \tilde{e}_k)$ ;

**end**

**end**

*% determine the posterior model probabilities;*  
**for**  $k \leftarrow 1$  **to**  $K$  **do**

$$\Pr(\mathcal{M} = k | \mathbf{y}) = \frac{p_k \cdot \tilde{e}_k}{\sum_{l=1}^K p_l \cdot \tilde{e}_l \cdot \exp[m(k) - m(l)]}$$

*% Compute the parameter estimates by PM;*  
 $\hat{\boldsymbol{\theta}}_k = \text{PM}(\boldsymbol{\theta}_k)$ ;

**end**

---

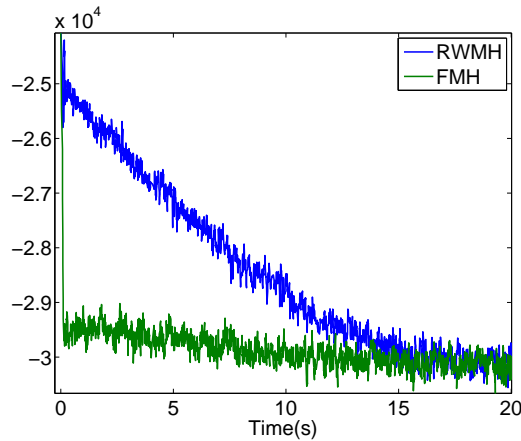


Figure 5.2: NLL evolution – posterior sampling with the two versions of the MH algorithm. On the abscissa it is represented the time in seconds. The two chains have the same initialization. The burn-in stage is considerably longer for the RWMH as compared to FMH.

1. The first study compares the two MH samplers: the standard RWMH and the FMH.
2. The second set of tests compares the two evidence approximations based on posterior samples, *i.e.*, the HMA and LMA.
3. The evaluation of our CEAPS classifier represents the third tests set. The classification performances for the CEAPS are first presented for various PSDs and then compared to those of the GMLE classifier, in a simplified scenario.
4. Results concerning the deconvolution are given in a visual form, by presenting the original textures, the observations and the deconvolved images.

#### 5.4.1 RWMH vs FMH

The first set of tests investigates the speed performances of two sampling algorithms, the isotropic RWMH and our efficient version FMH, in the context of the complicated laws for  $\theta$ .

Our tests indicate that the use of the FMH yields an algorithmic speed increase by a factor of at least 10 as compared to the RWMH. This is due to the FMH directional form of the proposal, which permits the algorithm to attain the high probability region in a very small number of iterations. Once in this region, the directional component will have negligible values and the algorithm will explore this high probability area of the parameter space due to the stochastic component of its proposal. This translates into a very short burn-in period, as opposed to the isotropic RWHM, which has a significantly longer burn-in period, depending on the initialization. This efficiency is illustrated in Figure 5.2 where the NLL chains for the two samplers are represented.

### 5.4.2 HMA and LMA

The posterior samples obtained using the FMH within Gibbs sampler are used to compute the evidences. This is achieved via two approximations: the PM-based LMA and the HMA.

These approximations are computed using the same set of posterior samples in order to evaluate their accuracy in the same conditions. The numerical results show that the difference between the two evidence approximations is less than 0.1%, thus confirming that both approximations are viable for the problem in question.

Moreover, since the sampling itself is the most costly part of the evidence computation, the choice of approximation does not affect the overall speed performance. Consequently, the two computational methods for our CEAPS demand roughly the same amount of time.

In the tests presented in the following section, the CEAPS is based on the HMA.

### 5.4.3 CEAPS performances

Let us now present the performances of the selection method itself. The experimental setup consists in testing our method on synthetic textures, using 20 sets of parameter values for the PSD. Each set was used for each PSD model to generate both GRF and SMGRF texture realizations. The observations are obtained in a scenario with Gaussian blur of standard deviation  $w = 0.3$  and SNR = 20dB. This corresponds to a partial overlap configuration of the texture PSD and the TF.

#### 5.4.3.1 CEAPS

The algorithm was run on each texture realization and Tables 5.2 and 5.3 summarize the classification results for GRF and SMGRF textures, respectively. We observe on the main diagonal of both tables the percentages of correct classifications. As expected, the CEAPS chooses the correct PSD model in most cases. There are, however, situations where the method chooses another model.

True model	Estimated model			
	Lo	GL	Exp	GG
Lo	85	10	1	4
GL	21	69	3	7
La	2	4	87	7
GG	4	8	16	72

Table 5.2: CEAPS model selection performance for GRF textures (correct classifications rate in %) for a partial overlap case.

As anticipated by the Fisher information analysis, in the majority of cases, there is enough available information on the central frequencies to ensure their accurate estimation.

True model	Estimated model			
	Lo	GL	Exp	GG
Lo	87	9	1	3
GL	<u>19</u>	73	2	6
La	1	7	83	9
GG	3	6	<u>20</u>	71

Table 5.3: CEAPS model selection performance for SMGRF textures (correct classifications rate in %) for a partial overlap case.

Nevertheless, the information concerning the widths is more sensitive to the noise level and the PSD model and thus more prone to estimation errors. These errors are important for the method functioning, since they trigger misclassifications. The majority of cases where the CEAPS fails are due to high noise levels and consist in mistakenly considering a PSD with thicker tails as the most likely model. In this situation, the thicker tails account for the noise and the noise level is underestimated.

Nevertheless, in the context of our model choice problem, where the nested models help testing the method’s ability to penalize model complexity, choosing another model can be regarded as not necessarily a failure. In this setting, the underlined percentages from Tables 5.2 and 5.3 represent the ”good” miss-classifications, for instance, a Generalized Gaussian with  $q = 1$  that is classified as an exponential. This illustrates the method’s capacity to penalize model dimension, *i.e.*, eliminate the parameters that do not significantly increase the model fit.

The method is not only able to distinguish between the different PSD forms, but also between the laws for the Fourier coefficients. More specifically, in 82% of the cases the algorithm correctly determined if the texture was from the GRF or the SMGRF class. This means that, on the one hand, the method has the ability to discriminate among a GRF and a SMGRF having the same form for the parametric part of the PSD. On the other hand, the PSD models themselves are structured enough to allow the algorithm to simultaneously identify the PSD model and whether all the PSD coefficients are identically scaled or not.

#### 5.4.3.2 CEAPS vs GMLE

A crucial point is that the GMLE cannot solve the problem of interest. This is due to the presence of indirect data introduced by the:

- non-Gaussian texture model,
- blurred and noisy observations.

Although the comparison cannot be performed on our problem, it is done on a simplified version of the problem consisting in direct observations (no noise and no convolution) of Gaussian textures.

Table 5.4 lists the average classification success rate for the CEAPS and the GMLE, every method being tested on 20 texture realizations, with various parameter values for

Algorithm	Classification accuracy (%)
CEAPS	89
GMLE	86

Table 5.4: Average model selection performance (classification success rate in %) comparison between CEAPS and GMLE. The averaging is done over PSD model and PSD parameters.

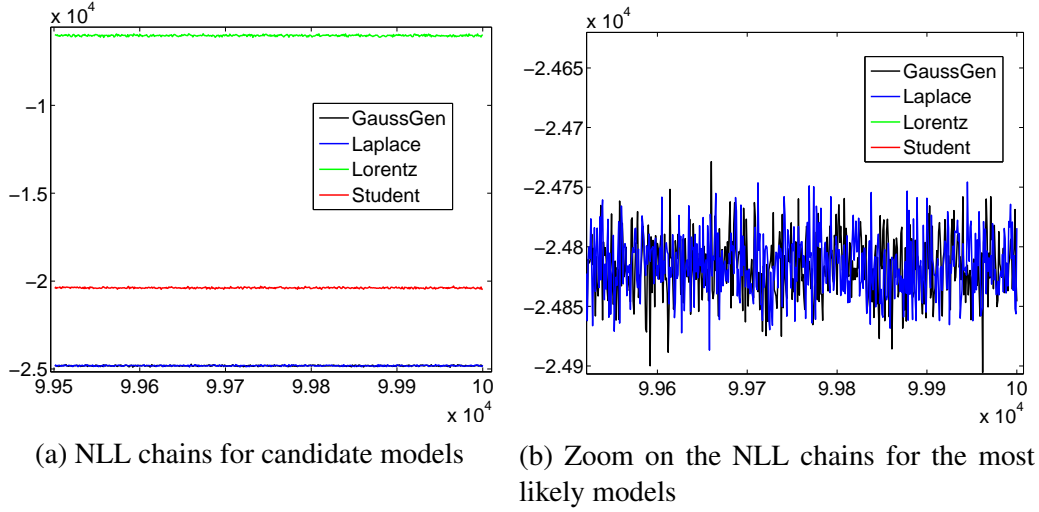


Figure 5.3: NLL chains to illustrate a typical situation where GMLE fails to select the good model

each type of PSD shape. More specifically, this represents an averaging over the PSD models and PSD parameters. The lower classification performance of the GMLE is due to the fact that it does not have any mechanism of model complexity penalization, thus it chooses the most complex among the embedded models. On the contrary, as previously explained, the CEAPS penalizes model dimension and selects the less complex model that fits the data.

Table 5.4 shows that the GMLE has a lower success rate and the reason for this is illustrated in Fig. 5.3. In this figure, we plotted a case where GMLE selects the Generalized Gaussian model, since its minimum NLL is the global NLL minimum among all models. However, the minimum neg-log PM is that of the Laplacian model, which is indeed the true model, this being a typical failure situation for GMLE.

As already stated in Section 5.1, the evidence based classifier is optimal from the risk point of view, which can be seen in this table through the CEAPS performances.

#### 5.4.4 Visual reconstructions

The experiments show that situations with high noise,  $\text{SNR} < 20\text{dB}$ , are challenging since the samples for the widths have a too strong variance. Furthermore, these samples,

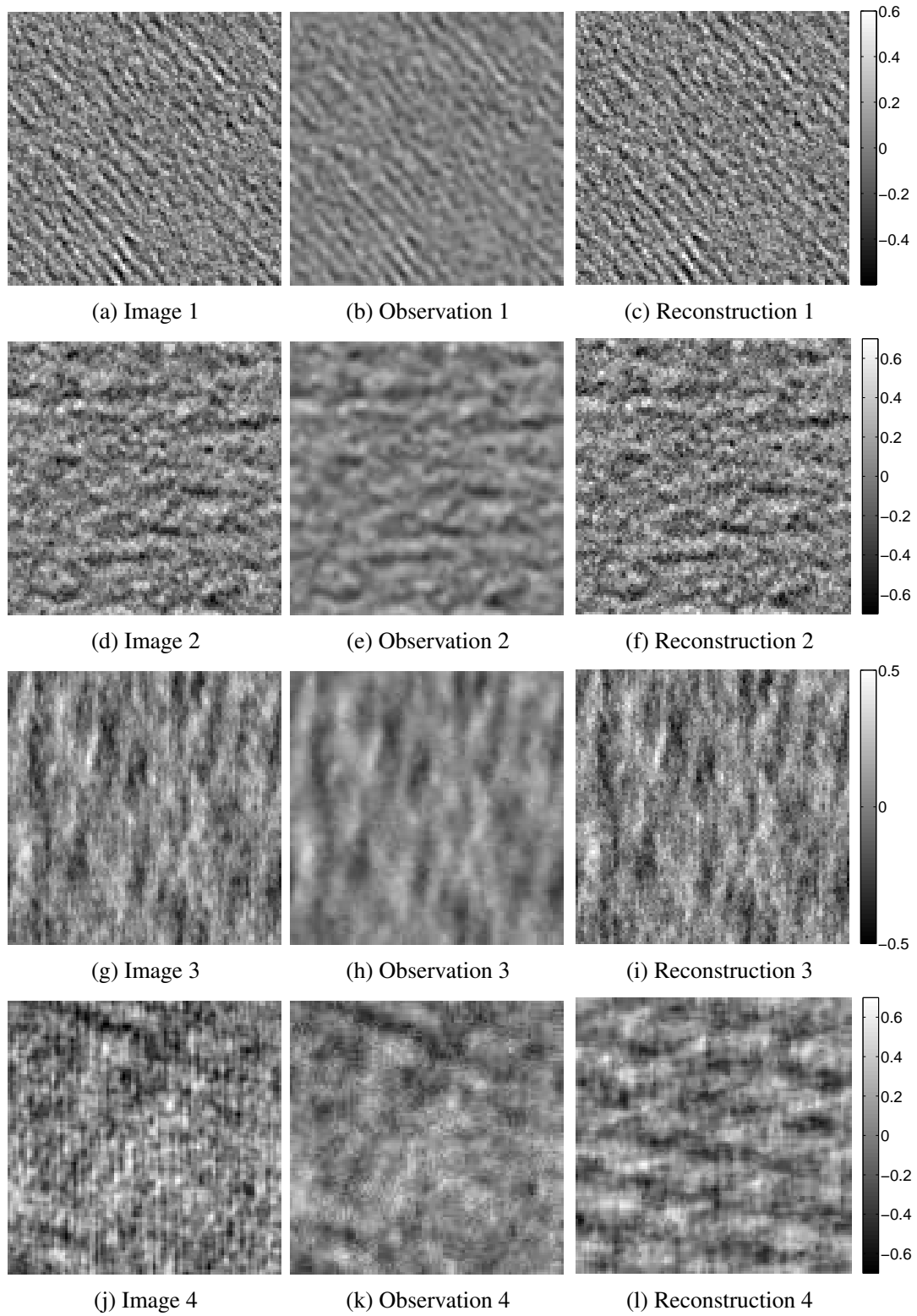


Figure 5.4: Reconstruction results: 4 cases to be read from left to right. On the first column the original, unobserved texture,  $x|\theta^*, \mathcal{M} = k^*$ , in the center the distorted observations,  $y$ , and on the right column the results of the deconvolution, for the selected model,  $\hat{x}|\hat{\theta}, \mathcal{M} = \hat{k}$ .

used to compute the evidences, have a direct impact on the model selection process.

In practice, above a certain level of noise, the method tends to favor the PSD shapes with thicker tails, by considering that these thicker tails account for the noise. In fact, for high noise levels there is a smaller amount of information, thus more uncertainty in the estimation, which eventually triggers estimation errors for the PSD widths and even misclassifications. More specifically, the noise level is underestimated and either the PSD widths are overestimated, or a model with thicker tails is selected.

Using the samples employed to compute the evidences, we can also compute PM estimates for the texture parameters, the noise precision and the unknown image. Consequently, as an additional result, our algorithm provides a PM estimate of the original image, conditionally on the selected model. Fig. 5.4 shows examples of the reconstruction. We can observe situations (Figs. 5.4c, 5.4f and 5.4i) where CEAPS successfully restores the texture even if the observations are severely degraded. This illustrates the method's high capacity to handle the blur and the noise. This is due to the strength of the information given by the structure of the PSD and to the method's optimality from the classification and estimation risk point of view. Nevertheless, there are also situations, such as Fig. 5.4l, where the image is degraded to an extent that impairs a reconstruction, in most cases, this being due to a low information scenario.

## 5.5 Conclusion and Perspectives

In this chapter we have presented a method for texture model choice from indirect observations. The textured images are modeled by Scale Mixture of Gaussian Random Fields or Gaussian Random Fields with parametric Power Spectral Density. By applying a Bayesian formalism, we were able to determine the posterior model probabilities based on the evidences, this approach being optimal from the risk point of view. Moreover, the within-model simulation technique that is employed translates into a sweep of all possible models and the computation of the evidence for each model in the dictionary. This quantity can be determined only by numeric methods, since the integral in (5.2) is intractable. We have compared several methods for numerically computing the evidence based on samples from the *a posteriori* law and we have presented the performances of these methods, deciding on which is the most adapted method for our application.

As a secondary result, this approach provides chains of samples for the parameters, conditionally on each model  $\mathcal{M} = k$ . These samples can be used to obtain estimates that are optimal from the mean square error point of view, by using the Posterior Mean estimator, the estimator that minimizes this error among all possible estimators.

The immediate perspective of this work is to compare our approach to other concurrent model choice methods.

Further developments include, but are not limited to, including the texture modeling related perspectives presented in Chapter 3, *i.e.*, extending the texture model to dependent Fourier coefficients, and the class of compared PSD models. However, that will come at a cost, *i.e.*, the coefficients dependence will be reflected in the computational complexity. Another idea is the use of multi-modal PSDs, in order to obtain more structured, quasi-



periodic textures.

From a different perspective, related to the work presented in Chapter 4 and [VGB14], the method can be adapted to deal with unknown non-parametric PSFs (blind) or parametric PSFs (semi-blind) and the estimation of their parameters.



# Deconvolution Segmentation for Textured Images

## Contents

<b>6.1</b>	<b>State of the art</b>	<b>73</b>
<b>6.2</b>	<b>Problem Statement</b>	<b>75</b>
<b>6.3</b>	<b>Bayesian Formulation</b>	<b>77</b>
6.3.1	Estimators	78
6.3.2	Computing the Estimators – <i>A Posteriori</i> Conditionals	79
<b>6.4</b>	<b>Sampling Aspects</b>	<b>80</b>
6.4.1	Sampling the Labels	80
6.4.2	Sampling the Image	82
<b>6.5</b>	<b>Results</b>	<b>87</b>
6.5.1	Evaluation of the exact method	89
6.5.2	Influence of the $\beta$ parameter	96
6.5.3	Influence of the approximations	97
<b>6.6</b>	<b>Conclusion and Perspectives</b>	<b>98</b>

The most complex problem in this thesis is the one of textured image segmentation from indirect observations. Most existing approaches do not tackle the indirect observations issue and focus only on segmentation. Image segmentation is a computer vision problem consisting in partitioning an image into several groups of adjacent pixels that have a certain homogeneity property (gray level, color, texture, or other features) or that compose an object of interest.

## 6.1 State of the art

The literature in the field of image segmentation is extensive as this topic has been of great interest for decades. Classifying the existing approaches is not an easy task, however, from the beginning, a distinction can be made between the **region-based** and the **contour-based** methods. An important aspect related to the region-based approaches is that they always provide closed contours, as opposed to the contour-based methods, for which the contours are not necessarily closed.

The most straightforward image segmentation method is thresholding, however, this method is seldom applicable, since it is only adapted for piecewise constant images. In this context, a large variety of approaches are defined for segmentation.

**Region growing** methods are iterative algorithms that start from initial seeds and expand each region in its neighborhood based on a similarity criterion. [ZZC10] presents a seeded image segmentation based on a heat diffusion process, [GUSV<sup>+</sup>09] describes an unsupervised region growing and multiresolution merging algorithm for color image segmentation, based on color edges and [AGBB12] presents a bottom-up aggregation image segmentation.

**Partial differential equations** based techniques have also been employed for image segmentation, for instance in [CV99] introducing an active contour without edges method for object segmentation, based on level sets, curve evolution and the Mumford Shah model.

A very popular approach to image segmentation is **graph partitioning**, where each pixel of the image is considered as the node of a graph and the goal is to connect the nodes having certain similarity properties. [FH04] uses a graph based image model and measures the evidence for a boundary between two regions in order to achieve an image segmentation. [BFL06] describes the basic framework for efficient object extraction from multi-dimensional image data using graph cuts. Also based on graph cuts, [BZ05] presents an image segmentation method based on a generalized Swendsen-Wang sampler. This method uses a Potts prior for the labels and, based on an adjacency graph, computes probabilities for each edge, performs graph clustering and graph flipping (instead of single vertex flipping as in the case of the Gibbs sampler).

The **watershed** approach is based on the topography of the gradient magnitude on the image. [MBLS01] presents a normalized cuts approach relying on a local measure of similarity of the textural characteristics in a neighborhood of the pixel, while [Gra06] uses a small number of predefined labels and computes for each unlabeled pixel a probability based on the speed of a random walk to reach a pre-labeled pixel. The final label assigned to a pixel is the one maximizing this probability. [SG07] unifies the graph cuts and the random walker methods in a common framework, based on  $L_q$  norms minimization for seeded image segmentation.

One of the first approaches for **textured image segmentation** is presented in [Tuc94], which is based on using the moments in small windows of the image as texture features. The topic of image segmentation for textured images has been previously addressed in [AG03]. This method consists in computing features based on the Discrete Wavelet Transform coefficients of blocks of the image, evaluating the difference between these features on adjacent blocks and applying post-processing techniques in order to obtain a one-pixel thick contours. This method does not provide a label field, thus no information about which texture belongs to which region. Another method providing texture edges [WHMM06] uses active contours and the patch based approach for texture analysis. However, none of the aforementioned approaches is formulated in the context of indirect observations.

Image segmentation for textured images is achieved in [LMS07] based on features extracted from the Fourier transform of the learning textures. A significant method for image segmentation based on both gray level (intervening contour framework) and texture (textons) is presented in [MBLS01]. The segmented image is obtained using a normalized cuts approach. [MRY<sup>+</sup>11] presents a method that models a homogeneous textured region of an image by a Gaussian distribution and the region boundaries by adaptive chain codes.

The segmentation is obtained using a clustering process. An image segmentation approach devoted to strongly resembling textures is given in [GSBB03]. The goal is to accurately characterize the textures and this is achieved by combining a collection of statistics and filter responses. This local information is then used in an aggregation process to determine the segmentation. A three stage segmentation method is presented in [LW06] and relies on characterizing both textured and non-textured regions using local spectral histograms. Texel based image segmentation is achieved in [TA09] by identifying the modes in the pdf of region properties. Nevertheless, this method does not deal with the aspect of indirect observations either.

A very significant class of segmentation methods are those relying on a **probabilistic** formulation and using models for the image and the labels. [GGGD90] presents an approach for image partitioning into homogeneous regions and for locating edges based on disparity measures. In [TZS01], an image segmentation method is developed based on MCMC and the  $K$  adventurers algorithm with the goal of preserving ambiguities in the segmentation process. This method integrates clustering and edge detection in the proposal probabilities. [DC04] introduced a weighted MRF model that estimates the model parameters and thus performs unsupervised image segmentation. [Mig06] formulates a Bayesian approach to image deconvolution with a regularization term based on a segmentation map, estimated using a Potts prior for the labels and a Gaussian model for the image, conditionally on the labels.

One of the most commonly used model for the labels in the probabilistic approaches is the Potts model. [CFP02] explores the Potts model based image segmentation by introducing a site dependent external field in the Potts model, while [PDBT13] proposes a method for jointly estimating the temperature parameter  $\beta$  of the Potts label prior using a likelihood free MH algorithm.

A very interesting work is the Bayesian method for image segmentation from indirect data (with known blur and Gaussian noise) in [AMD10]. This method is based on a Potts model for the labels, the pixels that belong to different regions being considered independent of each other. Within a region, the pixels are either independently Gaussian, or Markov-Gaussian. This type of image model makes the method suited mostly for the piecewise constant images. Nevertheless, the problem formulation in the Markov-Gaussian case is rather complicated and slightly unclear, resulting a difficult comprehension of the topic.

This chapter addresses a similar problem to the one in [AMD10] and presents a probabilistic method for image deconvolution and segmentation, based on a Potts model for the labels. However, in our case, the pixels of each region are modeled by a GRF. This allows us to segment images containing different textures, with the same gray level, but different textural characteristics.

## 6.2 Problem Statement

The current chapter presents a segmentation method for textured images affected by blur and noise. To be more exact,  $y$  is the blurred and noisy observation of the original

image  $\mathbf{x}$ . The unobserved image  $\mathbf{x}$  is composed of several regions, each of these regions consisting in a different texture. In our case, all the textures have the same mean gray level. This means that the information regarding the gray level in a neighborhood of a pixel is not pertinent in distinguishing the regions. In this context, the goal is to jointly segment and restore the image based on the textural characteristics.

To summarize, in this problem:

- the original image  $\mathbf{x}$ 
  - contains several regions,
  - each region consists in a patch of texture belonging to one of  $K$  classes,
  - each of the  $K$  textures is modeled by a GRF with parametric PSD. The parametric PSD has a known model driven by the unknown parameter set  $\boldsymbol{\theta}$ ;
- the PSF is known;
- the hyperparameters ( $\gamma_n$  – noise parameter, and  $\gamma_k$  – scale parameters for the textures) are unknown.

Let us consider that each patch of texture is extracted from a full image  $\mathbf{x}_k$  following the GRF model from Chapter 3, driven by the parameters  $\boldsymbol{\theta}_k$  and  $\gamma_k$ . Then, the variable hierarchy of our segmentation problem from indirect data is graphically represented in Figure 6.1:

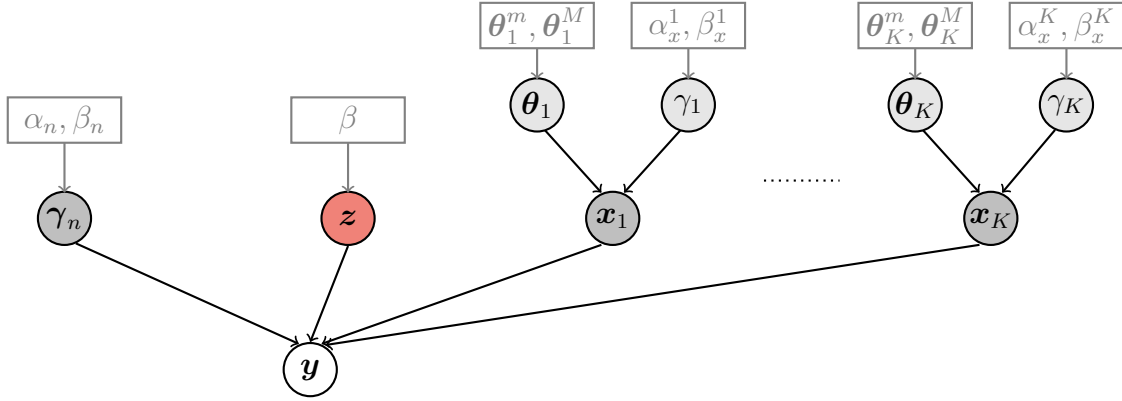


Figure 6.1: Hierarchical variable dependency for the deconvolution-segmentation of textured images.

Based on this variable dependency, the joint law for this problem can be expressed as:

$$f(\mathbf{y}, \mathbf{z}, \mathbf{x}_{1..K}, \gamma_n, \gamma_{1..K}, \boldsymbol{\theta}_{1..K}) = f(\mathbf{y}|\gamma_n, \mathbf{z}, \mathbf{x}_{1..K}) \cdot \pi(\gamma_n) \cdot f(\mathbf{z}|\beta) \cdot \prod_{k=1}^K f(\mathbf{x}_k|\boldsymbol{\theta}_k, \gamma_k) \cdot \prod_{k=1}^K \pi(\boldsymbol{\theta}_k) \cdot \prod_{k=1}^K \pi(\gamma_k) \quad (6.1)$$

As can be noticed in the variable dependency and in the joint law, the image  $\mathbf{x}$  is not directly probabilized and is not assigned a prior. Nevertheless, this variable represents a deterministic transformation of a series of probabilistic quantities. The process of obtaining the image  $\mathbf{x}$  containing the  $K$  textures, starting from the full textures  $\mathbf{x}_k$ ,  $k = 1..K$

and the labels  $\mathbf{z}$ , can be mathematically formalized as follows:

$$\mathbf{x} = \mathbf{S}_1 \mathbf{x}_1 + \mathbf{S}_2 \mathbf{x}_2 + \dots + \mathbf{S}_K \mathbf{x}_K = \sum_{k=1}^K \mathbf{S}_k \mathbf{x}_k \quad (6.2)$$

where  $\mathbf{S}_k = \mathbf{S}_k(\mathbf{z})$  are  $P \times P$  size diagonal binary matrices obtained based on the labels  $\mathbf{z}$ . These matrices extract from the single texture image  $\mathbf{x}_k$  the pixels with label  $k$  and replace the other pixels with 0. A more detailed description of this extraction and zero-padding process is given in Appendix D. The structure of the aforementioned matrices is:

$$\mathbf{S}_k = \text{Diag} \{ \delta(z_p, k), p = 1..P \}$$

where the entries equal to 1 are on the positions  $p$  with  $z_p = k$ .

This process can be illustrated by the schematic:

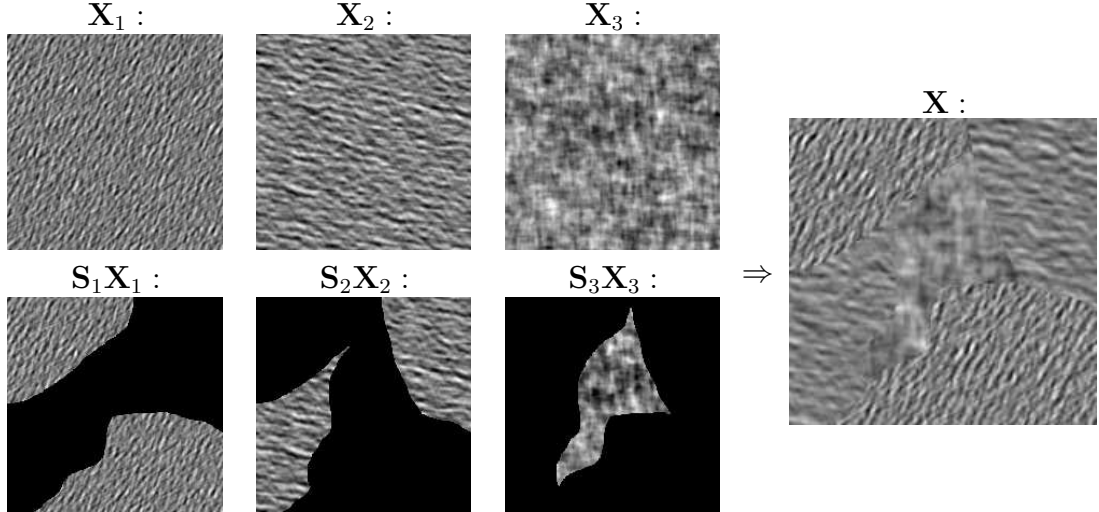


Figure 6.2: Image forming process based on the  $\mathbf{x}_k$  textures and the labels  $\mathbf{z}$ .

## 6.3 Bayesian Formulation

By using the form in (6.2) for the original image in (2.3), we obtain the following likelihood:

$$f(\mathbf{y}|\mathbf{z}, \mathbf{x}_{1..K}, \gamma_n) = (2\pi)^{-P} \cdot \gamma_n^P \cdot \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k \mathbf{x}_k \right\|^2 \right] \quad (6.3)$$

where  $\mathbf{H}$  represents the circulant, known convolution matrix. We have chosen a parametric form for this filter, driven by the width parameter  $w_f$ :

$$\circ h_{nm} = \exp \left[ -\frac{1}{2} w_f^2 (\nu_n^2 + \nu_m^2) \right] \quad (6.4)$$

This is an isotropic Gaussian filter, centered in the 0 frequency, and  $w_f$  is the inverse width of the Fourier transform of the convolution filter, as already employed in Chapters 4 and 5.

A prior must be assigned to the hidden labels, which encodes all the information we possess about this hidden field. An appropriate choice is the Potts model, which captures the tendency of neighbor labels of having the same value. The strength of this tendency is parametrized by the inverse temperature parameter  $\beta$ :

$$f(\mathbf{z}|\beta) = \mathcal{C}_z(\beta) \cdot \exp \left[ \beta \sum_{r \sim s} \delta(z_r, z_s) \right] \quad (6.5)$$

Further details about this model are given in Appendix C.

As for the rest of the unknowns, for the textures  $\mathbf{x}_k$  we have employed the GRF model, previously described in Section 3.2.1 and used in Chapter 4. The  $\gamma_n$  and  $\gamma_k$  have been assigned Gamma priors, while for the shape parameters of the textures,  $\boldsymbol{\theta}_k$ , we have chosen uniform priors.

Based on the likelihood law and the aforementioned priors, the joint law describing this problem (6.1) becomes:

$$\begin{aligned} f(\mathbf{y}, \mathbf{z}, \mathbf{x}_{1...K}, \gamma_n, \gamma_{1...K}, \boldsymbol{\theta}_{1...K}) = & \mathcal{C}_y \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k \mathbf{x}_k \right\|^2 \right] \\ & \cdot \mathcal{C}_{\gamma_n} \gamma_n^{\alpha_n + P - 1} \exp(-\gamma_n \beta_n) \cdot \mathcal{C}_z(\beta) \cdot \exp \left[ \beta \sum_{r \sim s} \delta(z_r, z_s) \right] \\ & \cdot \mathcal{C}_x \prod_k \left[ \left| \mathbf{R}_k(\boldsymbol{\theta}_k) \right|^{-1} \exp \left( -\gamma_k \left\| \mathbf{x}_k \right\|_{\mathbf{R}_k(\boldsymbol{\theta}_k)}^2 \right) \right] \\ & \cdot \prod_k \left[ \mathcal{C}_{\gamma_k} \cdot \gamma_k^{\alpha_k + P - 1} \cdot \exp(-\gamma_k \beta_k) \right] \cdot \prod_k \mathcal{U}_{\boldsymbol{\theta}_{km}, \boldsymbol{\theta}_{kM}}(\boldsymbol{\theta}_k) \end{aligned} \quad (6.6)$$

The *a posteriori* law, which is proportional to the joint law in (6.6), summarizes all the information about the unknowns contained by the data and the *a priori* models.

### 6.3.1 Estimators

Based on the aforementioned *a posteriori* law, parameter estimates can be obtained. The estimators we have chosen to use differ among the parameters:

- for the labels  $\mathbf{z}$  we have chosen a **marginalized MAP (mMAP)** estimator;
- for  $\gamma_n$ ,  $\gamma_k$  and  $\boldsymbol{\theta}_k$  we have used the **PM** estimator;
- for the textures  $\mathbf{x}_k$  we have used the **PM** estimator, conditionally on the estimated labels  $\hat{\mathbf{z}}$ .



An estimate of the original image  $\mathbf{x}$  can be obtained based on the estimates for the labels  $\hat{\mathbf{z}}$  and for the textures  $\hat{\mathbf{x}}_k$  as follows:

$$\hat{\mathbf{x}} = \hat{\mathbf{S}}_1 \hat{\mathbf{x}}_1 + \hat{\mathbf{S}}_2 \hat{\mathbf{x}}_2 + \dots + \hat{\mathbf{S}}_K \hat{\mathbf{x}}_K \quad (6.7)$$

where the estimated extraction matrices  $\hat{\mathbf{S}}_k = \hat{\mathbf{S}}_k(\hat{\mathbf{z}})$  are obtained based on  $\hat{\mathbf{z}}$ .

### 6.3.2 Computing the Estimators – *A Posteriori* Conditionals

The complicated posterior law cannot be exploited directly to straightforwardly determine the estimates. Consequently, the information must be extracted using numerical methods. The Gibbs sampler is employed in this work. This algorithm sequentially samples the *a posteriori* conditional laws, which in this case write:

- the noise parameter  $\gamma_n$  has a standard Gamma conditional, meaning that its sampling does not pose any difficulties;

$$f(\gamma_n | *) \propto \gamma_n^{\alpha_n + P - 1} \cdot \exp \left[ -\gamma_n \left( \left\| \mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k \mathbf{x}_k \right\|^2 + \beta_n \right) \right] \quad (6.8)$$

- the PSD scale parameters  $\gamma_k$ ,  $k = 1..K$ , also have Gamma forms, which can be straightforwardly sampled;

$$f(\gamma_k | *) \propto \gamma_k^{\alpha_k + P - 1} \cdot \exp \left[ -\gamma_k \left( \left\| \mathbf{x}_k \right\|_{\mathbf{R}_k(\boldsymbol{\theta}_k)}^2 + \beta_k \right) \right] \quad (6.9)$$

- the conditional law of the PSD shape parameters  $\boldsymbol{\theta}_k$ ,  $k = 1..K$ , is highly non-linear and has a non-standard form. Nevertheless, this law can be sampled using an RWMH step;

$$f(\boldsymbol{\theta}_k | *) \propto \prod_p \lambda_p(\boldsymbol{\theta}_k) \cdot \exp \left[ -\gamma_k \sum_p \lambda_p(\boldsymbol{\theta}_k) |x_{kp}|^2 \right] \cdot \mathcal{U}_{\theta_{km}, \theta_{kM}}(\boldsymbol{\theta}_k) \quad (6.10)$$

- the labels have a non-standard complicated law. The sampling of this law will be detailed in the next section.

$$f(\mathbf{z} | *) \propto \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k(\mathbf{z}) \mathbf{x}_k \right\|^2 + \beta \sum_{r \sim s} \delta(z_r, z_s) \right] \quad (6.11)$$

- the law for each complete texture  $\mathbf{x}_k$ ,  $k = 1..K$ , has a Gaussian, non-separable form, of dimension  $P$ . The sampling of this law is explained in a subsequent section;

$$f(\mathbf{x}_k | *) \propto \exp \left\{ -\frac{1}{2} \left[ \gamma_n \left\| \mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k \mathbf{x}_k \right\|^2 + \gamma_k \left\| \mathbf{x}_k \right\|_{\mathbf{R}_k(\boldsymbol{\theta}_k)}^2 \right] \right\} \quad (6.12)$$

Based on the values of the samples drawn from the posterior law via the Gibbs algorithm, the estimators are computed as follows:

- The mMAP labels estimation consists in selecting for each label the value that has been most often selected during the sampling process. In practice, this comes down to independently computing an *a posteriori* histogram of the sampled values for each label and selecting the value that corresponds to the maximum in the histogram. This value gives the estimate  $\hat{z}$ .
- The PM estimates of the rest of the variables consist in calculating the expectancy of the posterior law. Since this comes down to calculating an intractable integral, practically, the estimates are obtained numerically, by averaging the posterior samples.

## 6.4 Sampling Aspects

This section provides detailed explanations regarding the cumbersome task of sampling the labels, *i.e.*, the  $f(z|*)$  law in (6.11), and the full textures, *i.e.*, the  $f(x_k|*)$  law in (6.12). These two sampling processes represent the major algorithmic challenges of our approach. Since the treated problem has not been tackled in the literature before, these aspects had never been encountered and addressed, either. For this reason, the sampling processes, especially that of the textures, have been conceptualized in several different forms that have allowed us to gradually mature our perspective and arrive at the version presented in this section.

### 6.4.1 Sampling the Labels

The conditional law for the labels (6.11) has a complicated and non separable form and thus its sampling is not an easy task. A first solution is to apply Gibbs sampling and thus sample the labels sequentially, each conditionally on the rest of the labels. Nevertheless, this solution implies a prohibitive computational time, thus the goal is to parallelize this process.

The prior Potts law has been defined using a 4-neighborhood. This prior field can be sampled using a Gibbs algorithm in only two steps. In each of these steps, we are sampling labels that are conditionally independent of each other. In this context, the label lattice can be viewed as a checkerboard [Win06], where the "white" labels are independent of each other, conditionally on the "black" labels. The same reasoning applies to the "black" labels, which are mutually independent, conditionally on the "white" ones.

However, these considerations are only related to the prior. The *a posteriori* conditional law for the labels also contains the likelihood term. This term can be expressed in a separable form with respect to the labels, thus the label sampling can be performed in a parallel, computationally efficient manner.

**Remark:** If this were not true, the sampling should have been performed sequentially, one label at a time, conditionally on the rest of the labels and variables. This would have

translated into a prohibitively costly sampling process.

Sampling a label, say  $z_p$  comes down to computing all the probabilities:

$$\begin{aligned}
 \Pr(z_p = 1|*) &\propto \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H}\mathbf{x}_1^p \right\|^2 + \beta \sum_{r \sim p} \delta(z_r, 1) \right] \\
 &\dots \\
 \Pr(z_p = k|*) &\propto \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H}\mathbf{x}_k^p \right\|^2 + \beta \sum_{r \sim p} \delta(z_r, k) \right] \\
 &\dots \\
 \Pr(z_p = K|*) &\propto \exp \left[ -\gamma_n \left\| \mathbf{y} - \mathbf{H}\mathbf{x}_K^p \right\|^2 + \beta \sum_{r \sim p} \delta(z_r, K) \right]
 \end{aligned} \tag{6.13}$$

up to the multiplicative factor. This factor can be determined knowing that the sum of the probabilities is 1.

The  $\mathbf{x}_k^p$  in (6.13) represents an image that has all the pixels identical to  $\mathbf{x}$ , except for pixel  $p$ . The pixel at position  $p$  in  $\mathbf{x}_k^p$  is obtained by extracting the pixel  $p$  from  $\mathbf{x}_k$ . In this manner, the data adequacy term measures which class is the most appropriate for describing pixel  $p$  in the sense that it provides the maximum level of coherence with the rest of the image.

The second term is the contribution of the prior. This term is meant to ensure a coherence of the current label with its neighbors. This is weighted by the inverse temperature parameter,  $\beta$ , which tunes the strength of the connection with the neighbor labels.

To compute the probabilities in (6.13), we must evaluate the value of the two terms at pixel  $p$ . The second term is easily computed by counting the neighbors of pixel  $p$  having label  $k$  and multiplying with  $\beta$ .

Let us now focus on the first term, that we will denote  $V_{p,k} = \left\| \mathbf{y} - \mathbf{H}\mathbf{x}_k^p \right\|^2$ . In order to write this term in a more convenient form, let us introduce the following auxiliary quantities:

- a vector  $\mathbb{1}_p$  in which only one element is different from 0. This element is equal to 1 and has the index  $p$ :

$$\mathbb{1}_p = \underbrace{[0 \ 0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]}_P^t$$

- a scalar quantity  $\delta_{kp} \in \mathbb{C}$  that records the difference between the  $p$ th pixels of the full texture image  $\mathbf{x}_k$  and of the image  $\mathbf{x}$ :

$$\delta_{kp} = -\mathbf{x}_p + \mathbf{x}_{kp}$$

Then, using these notations,  $V_{p,k}$  can be written as:

$$\begin{aligned}
 V_{p,k} &= \left\| \mathbf{y} - \mathbf{H} (\mathbf{x} - \delta_{kp} \mathbf{1}_p) \right\|^2 \\
 &= \left\| \underbrace{(\mathbf{y} - \mathbf{H}\mathbf{x})}_{\bar{\mathbf{y}}} - \delta_{kp} \mathbf{H} \mathbf{1}_p \right\|^2 \\
 &= \bar{\mathbf{y}}^\dagger \bar{\mathbf{y}} + \delta_{kp}^2 \underbrace{\mathbf{1}_p^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{1}_p}_{\|\mathbf{H} \mathbf{1}_p\|^2} - 2\delta_{kp} \mathbf{1}_p^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}
 \end{aligned} \tag{6.14}$$

Let us analyze each of these terms:

1. The first term  $\bar{\mathbf{y}}^\dagger \bar{\mathbf{y}}$  does not depend on  $p$  or  $k$ . This means that its computation is not required in the process of label sampling, thus this term can be included in the multiplicative factor;
2. The  $\|\mathbf{H} \mathbf{1}_p\|^2$  factor of the second term does not depend on  $p$ . This is due to the circulant form of the  $\mathbf{H}$  matrix. For this reason, this norm only needs to be computed once for the entire problem. In fact, this norm amounts to the sum  $\sum_{p=1}^P |\hat{h}_p|^2$ . Consequently, we only need to compute the  $\delta_{kp}$ , which can be done in parallel for all values of  $p$  and  $k$  and stored in a  $N \times N \times K$  block.
3. Finally, the third term can be written as:

$$\delta_{kp} \mathbf{1}_p^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}} = \delta_{kp} \mathbf{1}_p^\dagger \underbrace{\mathbf{H}^\dagger \bar{\mathbf{y}}}_{\substack{\text{FFT} \\ \text{select}}} \tag{6.15}$$

The product  $\mathbf{H}^\dagger \bar{\mathbf{y}}$  can be computed efficiently by FFT, once for all the iterations. The product with  $\mathbf{1}_p^\dagger$  consists in selecting the pixel  $p$  of  $\mathbf{H}^\dagger \bar{\mathbf{y}}$ .

### 6.4.2 Sampling the Image

The law in (6.12) is Gaussian, however, its sampling will pose problems due to the high dimension of the variable. Let us rewrite this expression as:

$$f(\mathbf{x}_k | *) \propto \exp \left\{ -\frac{1}{2} \|\mathbf{x}_k - \mathbf{m}_k\|_{\Sigma_k}^2 \right\} \tag{6.16}$$

where

$$\begin{aligned}
 \Sigma_k &= \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1}(\boldsymbol{\theta}_k) \right)^{-1} = \mathbf{Q}_k^{-1} \\
 \mathbf{m}_k &= \Sigma_k \cdot \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}_k \\
 \bar{\mathbf{y}}_k &= \mathbf{y} - \mathbf{H} \sum_{l \neq k} \mathbf{S}_l \mathbf{x}_l
 \end{aligned} \tag{6.17}$$

This  $\bar{\mathbf{y}}_k$  represents in fact the observations from which we are subtracting the contribution of all the regions, except region  $k$ . In fact,  $\bar{\mathbf{y}}_k$  collects the part of the observations that are

relevant about texture  $k$ .

**Remark:** To improve the readability of the equations, in the following we will use the simplified notation  $\mathbf{R}_k = \mathbf{R}_k(\boldsymbol{\theta}_k)$ .

The  $P$ -dimensional Gaussian law in (6.16) has mean  $\mathbf{m}_k$  and covariance matrix  $\Sigma_k$ . The sampling of this high dimension Gaussian requires computing the matrix  $\Sigma_k$ , which is a very costly or even impossible operation due to the fact that it implies determining the inverse of a  $P \times P$  matrix (for instance, for a  $256 \times 256$  pixel image,  $\Sigma_k$  is of size  $65536 \times 65536$ ). This can be performed efficiently only in the case where the covariance matrix is sparse or has a special structure (circulant).

In our case, the  $\mathbf{R}_k$ ,  $\mathbf{H}$  and, by extension, the  $\mathbf{H}^\dagger \mathbf{H}$  matrices are circulant. However, the presence of the  $\mathbf{S}_k$  matrix breaks the circularity of the first term, making it impossible to perform the sampling straightforwardly and efficiently by FFT. Moreover, since this sampling operation must be performed at each iteration, it is obvious that a direct computation would render the algorithm prohibitively expensive.

Nevertheless, the literature accounts for a method of sampling high dimension Gaussians that have the covariance matrix with a certain structure [OFG12]. This method is called Sampling by Perturbation-Optimization and consists in building a "perturbed" criterion, based on the target law, and optimizing this criterion in order to obtain a sample of the target. The Sampling by Perturbation-Optimization can be applied in cases where the precision matrix and the mean can be written as a sum of the form:

$$\begin{aligned}\boldsymbol{\Pi} &= \sum_{c=1}^C \mathbf{M}_c^\dagger \mathbf{R}_c^{-1} \mathbf{M}_c \\ \boldsymbol{\mu} &= \boldsymbol{\Pi}^{-1} \sum_{c=1}^C \mathbf{M}_c^\dagger \mathbf{R}_c^{-1} \boldsymbol{\mu}_c\end{aligned}\tag{6.18}$$

**Remark:** This sampling method is only sure to provide samples of the target law if the optimization is perfect. The work in [GMI13] provides a means of ensuring that the obtained samples are indeed from the target even in the case of an imperfect optimization.

By identifying  $\boldsymbol{\Pi} = \Sigma_k^{-1}$ , the most obvious choice is  $C = 2$  and:

$$\begin{cases} \mathbf{M}_1^\dagger &= \mathbf{S}_k^\dagger \mathbf{H}^\dagger \\ \mathbf{R}_1 &= 1/\gamma_n \mathbf{I}_P \\ \boldsymbol{\mu}_1 &= \bar{\mathbf{y}}_k \end{cases} \quad \begin{cases} \mathbf{M}_2^\dagger &= \mathbf{I}_P \\ \mathbf{R}_2 &= 1/\gamma_k \mathbf{R}_k \\ \boldsymbol{\mu}_2 &= \mathbf{O}_P \end{cases}$$

The perturbation phase of this algorithm consists in drawing the following samples:

$$\begin{aligned}\boldsymbol{\xi}_1 &\sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{R}_1) \\ \boldsymbol{\xi}_2 &\sim \mathcal{N}(\boldsymbol{\mu}_2, \mathbf{R}_2)\end{aligned}\tag{6.19}$$

The cost of sampling these variables is not prohibitive:  $\boldsymbol{\xi}_1$  corresponds to a decorrelated white noise of mean  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\xi}_2$  corresponds to a texture having the same model as the prior for  $\mathbf{x}_k$ .  $\boldsymbol{\xi}_2$  can be easily obtained by FFT.

In order to obtain a sample of the textured image  $\mathbf{x}_k$ , the following criterion must be optimized:

$$\begin{aligned} J(\mathbf{x}_k | \boldsymbol{\xi}_1, \boldsymbol{\xi}_2) &= \gamma_n (\boldsymbol{\xi}_1 - \mathbf{H} \mathbf{S}_k \mathbf{x}_k)^\dagger (\boldsymbol{\xi}_1 - \mathbf{H} \mathbf{S}_k \mathbf{x}_k) + \gamma_k (\boldsymbol{\xi}_2 - \mathbf{x}_k)^\dagger \mathbf{R}_k^{-1} (\boldsymbol{\xi}_2 - \mathbf{x}_k) \\ &= \mathbf{x}_k^\dagger \mathbf{Q}_k \mathbf{x}_k - 2 \mathbf{x}_k^\dagger \mathbf{q}_k + \text{cte.} \end{aligned} \quad (6.20)$$

where  $\mathbf{q}_k = \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \mathbf{R}_k^{-1} \boldsymbol{\xi}_2$ .

### 6.4.2.1 Optimization algorithm

The advantage of this approach is that it does not need to store or compute the inverse of the matrix  $\mathbf{Q}_k$ , which is of size  $P \times P$ . We have focused on optimal step gradient-based algorithms. The techniques we are employing are all guaranteed to converge. The advantage of these approaches is that the only necessary ingredients for performing the optimization are the methods to compute the product  $\mathbf{Q}_k \mathbf{x}_k$  and  $\mathbf{q}_k$ .

- The product can be written as:

$$\mathbf{Q}_k \mathbf{x}_k = \gamma_n \underbrace{\mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k}_{\substack{\text{select} \\ \text{FFT} \\ \text{select}}} \mathbf{x}_k \quad (6.21)$$

and thus efficiently computed through a series of FFT and pixel selection. The selection process is achieved in the spatial domain, while the costly matrix multiplications are performed in the Fourier domain.

- The  $\mathbf{q}_k$  term writes:

$$\mathbf{q}_k = \gamma_n \underbrace{\mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1}_{\substack{\text{FFT} \\ \text{select}}} + \gamma_k \underbrace{\mathbf{R}_k^{-1} \boldsymbol{\xi}_2}_{\text{FFT}} \quad (6.22)$$

and can also be efficiently computed by FFT.

Theoretically, there is no constraint on the optimization technique to be used at this point. We have chosen the:

- gradient descent,
- conjugate gradient.

The high dimension of the problem seems to favor the conjugate gradient optimization, since it is faster than the classical gradient descent. However, we have experienced convergence difficulties, *i.e.*, the optimization algorithm did not converge in  $P$  iterations, making the overall algorithm very slow. In practice, we have noticed that the step length at each iteration was extremely low. For this reason, irrespectively of the initialization, the evolution in the parameter space was very slow. Consequently, the differences between the current values and the updates were almost insignificant. This represents the cause of the extremely reduced convergence speed.

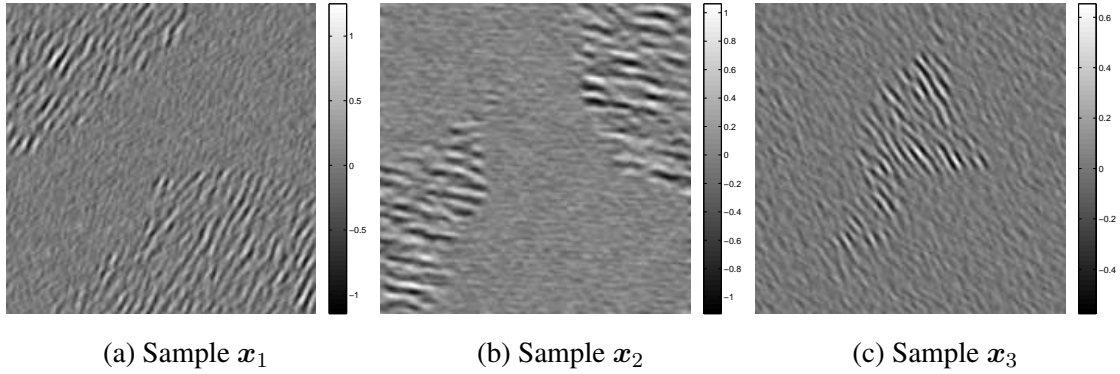


Figure 6.3: Samples of the full textures for a problem with  $K = 3$ .

The solution for this problem was the use of the preconditioner:

$$\mathbf{M} = \gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1} \quad (6.23)$$

This form for the preconditioner has not been randomly chosen. It is in fact an approximation of the inverse Hessian of the problem that we are trying to solve. This approximation has a Circulant-block-Circulant form and was used instead of the actual inverse Hessian, which has no special form, for computational efficiency reasons.

This preconditioner was used for both of the aforementioned optimization algorithms and we have compared the performances. Thusly, we have compared:

- a preconditioned gradient descent,
- a preconditioned conjugate gradient.

In this context, the two methods have yielded similar results, requiring only a few iterations to converge.

Finally, we have chosen to focus on the **preconditioned gradient descent** method. This algorithm requires at each iteration the computation of the descent direction. This is represented by the product  $\mathbf{M} \mathbf{g}_k^{(i)}$ , where  $\mathbf{g}_k^{(i)} = \mathbf{Q}_k \mathbf{x}_k^{(i)} - \mathbf{q}_k$  is the gradient at iteration  $i$ . Since the matrix  $\mathbf{M}$  is circulant, this product can be easily computed in the Fourier domain.

The second ingredient that is necessary is the step length at each iteration  $\alpha_i$ :

$$\alpha_i = \frac{\mathbf{g}_k^{(i)\dagger} \mathbf{M}^\dagger \mathbf{g}_k^{(i)}}{\mathbf{g}_k^{(i)\dagger} \mathbf{M}^\dagger \mathbf{Q}_k \mathbf{M} \mathbf{g}_k^{(i)}} \quad (6.24)$$

Figure 6.3 illustrates a set of full textures samples. A first remark that can be made is that these samples are not homogeneous. This is due to the data term for the texture  $x_k$ , which contains information only on the positions having label  $k$ . For the rest of the pixels, the only contribution comes from the regularization term.

#### 6.4.2.2 Approximating $\mathbf{Q}_k$

The most direct possibility to increase the computational efficiency in this case is to approximate  $\mathbf{Q}_k$  by a Circulant-block-Circulant matrix. This allows us to perform the

sampling by FFT and yields a considerable gain in terms of speed.

The exact solution of (6.20) is:

$$\tilde{\mathbf{x}}_k = \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1} \right)^{-1} \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \mathbf{R}_k^{-1} \boldsymbol{\xi}_2 \right) \quad (6.25)$$

### First circulant approximation

This approximation consists in eliminating the  $\mathbf{S}_k$  matrix from the expression of  $\mathbf{Q}_k$ , in order to keep only Circulant-block-Circulant matrices:

$$\dot{\mathbf{Q}}_k = \gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1} \quad (6.26)$$

The solution of the approximate system, corresponding to (6.25), is:

$$\dot{\mathbf{x}}_k = \left( \gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1} \right)^{-1} \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \mathbf{R}_k^{-1} \boldsymbol{\xi}_2 \right) \quad (6.27)$$

In this case, due to the Circulant-block-Circulant structure of the covariance matrix, the Fourier coefficients of  $\mathbf{x}_k$  are independent and thus can be sampled in parallel. Each of these coefficients, say  $\dot{x}_{kp}$  has the form:

$$\dot{x}_{kp} = \left[ \gamma_n |\dot{h}_p|^2 + \gamma_k \lambda_p(\boldsymbol{\theta}_k) \right]^{-1} \left( \gamma_n \dot{h}_p^\dagger \dot{\xi}_{1p} + \gamma_k \lambda_p(\boldsymbol{\theta}_k) \dot{\xi}_{2p} \right) \quad (6.28)$$

### Second circulant approximation

A second and more rude approximation for the matrix  $\mathbf{Q}_k$  is to completely eliminate the  $\mathbf{H}$ :

$$\ddot{\mathbf{Q}}_k = \gamma_n \mathbf{I} + \gamma_k \mathbf{R}_k^{-1} \quad (6.29)$$

The solution of the new approximate system is:

$$\ddot{\mathbf{x}}_k = \left( \gamma_n \mathbf{I} + \gamma_k \mathbf{R}_k^{-1} \right)^{-1} \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \mathbf{R}_k^{-1} \boldsymbol{\xi}_2 \right) \quad (6.30)$$

Similarly to the case of the first approximation, the covariance matrix has a Circulant-block-Circulant structure, *i.e.*, it is diagonalizable by FFT. Consequently, the Fourier coefficients of  $\mathbf{x}_k$  are independent and can be sampled in parallel. Each of these coefficients, say  $\ddot{x}_{kp}$ , has the form:

$$\ddot{x}_{kp} = \left[ \gamma_n + \gamma_k \lambda_p(\boldsymbol{\theta}_k) \right]^{-1} \left( \gamma_n \dot{h}_p^\dagger \dot{\xi}_{1p} + \gamma_k \lambda_p(\boldsymbol{\theta}_k) \dot{\xi}_{2p} \right) \quad (6.31)$$

**Remark:** This approximation does not increase the speed performance as compared to the first approximation, being in fact a special case of the latter.

### Cost of the approximations



The proposed approximations allow for a significant speed increase as compared to the "exact" method based on sampling by optimization. Nevertheless, this comes at a cost, since we are in fact modifying the criterion that must be optimized. Consequently, it is interesting to analyze the difference between the expected results. This can be done by comparing the expectation and the variance of the solutions for the three systems:

- Exact system

$$\begin{aligned}\mathbb{E} [\tilde{\mathbf{x}}_k] &= \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1} \right)^{-1} \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}_k \\ \text{var} [\tilde{\mathbf{x}}_k] &= \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1} \right)^{-1}\end{aligned}\tag{6.32}$$

- First approximation

$$\begin{aligned}\mathbb{E} [\dot{\mathbf{x}}_k] &= (\gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1})^{-1} \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}_k \\ \text{var} [\dot{\mathbf{x}}_k] &= (\gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1})^{-1} \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1} \right) \cdot (\gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1})^{-1}\end{aligned}$$

- Second approximation

$$\begin{aligned}\mathbb{E} [\ddot{\mathbf{x}}_k] &= (\gamma_n \mathbf{I} + \gamma_k \mathbf{R}_k^{-1})^{-1} \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}_k \\ \text{var} [\ddot{\mathbf{x}}_k] &= (\gamma_n \mathbf{I} + \gamma_k \mathbf{R}_k^{-1})^{-1} \left( \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \mathbf{R}_k^{-1} \right) \cdot (\gamma_n \mathbf{I} + \gamma_k \mathbf{R}_k^{-1})^{-1}\end{aligned}$$

By analyzing the statistics of these solutions, we can notice that they differ both in mean and in variance. From a theoretical point of view, this means that we are not actually sampling the same law. The following section will provide a comparison of the method performances in the three cases in order to evaluate the impact of the approximations on the estimation results.

## 6.5 Results

This chapter has presented so far our method for joint image segmentation and deconvolution with hyperparameters and texture parameters estimation. This is a very complex problem, which has confronted us with a series of algorithmic challenges, especially related to the sampling of the labels and of the textures. These challenges were overcome due to our adapted formulation of the problem and the resulting method provides not only an estimated label field, but also an estimate of the original image.

The problem of segmentation has a considerable degree of difficulty, especially in this case, where the regions consist in textures with the same mean and the data is affected by blur and noise.

The practical implementation of our deconvolution-segmentation method is described by Algorithm 3. Its implementation has lead us to a series of practical considerations:

**Algorithm 3:** Deconvolution-Segmentation Algorithm for Textured Images

---

**input** : Data  $\mathbf{y}$ , number of textures  $K$ , temperature parameter for the Potts field  $\beta$   
**output**: Samples for the labels  $\mathbf{z}^{(t)}$ , the textures  $\mathbf{x}_k^{(t)}$ , noise parameter  $\gamma_n^{(t)}$ , texture parameters  $\boldsymbol{\theta}_k^{(t)}$  and  $\gamma_k^{(t)}$  ( $t = 1..T$ )

*% Gibbs sampler for  $(\mathbf{x}_{1..K}, \gamma_{1..K}, \boldsymbol{\theta}_{1..K}, \gamma_n, \mathbf{z})$*

initializations  $t = 0$ ,  
 $\gamma_n^{(0)}, \boldsymbol{\theta}_{1..K}^{(0)}, \gamma_{1..K}^{(0)}$ ,  
 $\mathbf{x}^{(0)} = \mathbf{y}$ ,  
 $\mathbf{z}^{(0)} = \text{ceil}(K * \text{rand}(N))$ ;

**while not convergence do**  
   $t = t + 1$ ;  
  **for**  $k = 1$  **to**  $K$  **do**  
     $\mathbf{x}_k^{(t)} = \text{SampleTexture}(\mathbf{y}, \mathbf{z}^{(t-1)}, \gamma_n^{(t-1)}, \mathbf{x}^{(t-1)}, \gamma_k^{(t-1)}, \boldsymbol{\theta}_k^{(t-1)})$ ;  
     $\gamma_k^{(t)} \sim f(\gamma_k | \mathbf{x}_k^{(t)}, \boldsymbol{\theta}_k^{(t-1)}, \alpha_x, \beta_x)$ ;  
     $\boldsymbol{\theta}_k^{(t)} \sim \text{RWMH with target } f(\boldsymbol{\theta}_k | \mathbf{x}_k^{(t)}, \gamma_k^{(t)})$ ;  
  **end**  
   $\gamma_n^{(t)} \sim f(\gamma_n | \mathbf{y}, \mathbf{x}_{1..K}^{(t)}, \mathbf{z}^{(t-1)}, \alpha_n, \beta_n)$ ;  
   $\mathbf{z}^{(t)} = \text{SampleLabels}(\mathbf{y}, \gamma_n^{(t)}, \mathbf{x}_{1..K}^{(t)}, \beta)$ ;  
  *% count label occurrences for every pixel p*  
   $\text{LabelOcc}(p, z_p) = \text{LabelOcc}(p, z_p) + 1$ ;  
**end**  
  *% determine the parameter estimates;*  
   $\hat{\gamma}_n = \text{PostMean}(\gamma_n^{(t)})$ ;  
  **for**  $k = 1$  **to**  $K$  **do**  
     $\hat{\mathbf{x}}_k = \text{PostMean}(\mathbf{x}_k^{(t)})$ ;  
     $\hat{\gamma}_k = \text{PostMean}(\gamma_k^{(t)})$ ;  
     $\hat{\boldsymbol{\theta}}_k = \text{PostMean}(\boldsymbol{\theta}_k^{(t)})$ ;  
  **end**  
  *% determine the final segmentation: most frequently sampled label for each pixel;*  
   $\hat{\mathbf{z}} = \text{MaxOccurrenceLabel}(\text{LabelOcc})$ ;  
  *% determine the reconstructed image;*  
   $\hat{\mathbf{x}} = \text{BuildImage}(\hat{\mathbf{z}}, \hat{\mathbf{x}}_{1..K})$ ;

---

- we propose to initialize the labels by a white noise with values in the range  $1..K$ . Our tests have shown that the algorithm converges faster when using this initialization as compared to an initialization by a constant label field;
- the use of the approximations for the  $\mathbf{Q}_k$  matrix in the texture sampling phase yields a speed acceleration by a factor of 5. Whilst the exact sampling method takes about 5 minutes for a  $200 \times 200$  pixel image, the method based on approximated sampling by FFT takes less than 1 minute;
- the preconditioned gradient descent and the preconditioned conjugate gradient methods have similar performances in this case. The non preconditioned versions of these algorithms are very slow to converge for this problem;
- using a convenient initialization for the textures in the optimization algorithm can significantly reduce the convergence time. We have tested several initializations ranging from a constant image to a white noise. Obviously, the best performances are obtained for an approximation of the sample:

$$\mathbf{x}_k^{init} = (\gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \mathbf{R}_k^{-1})^{-1} (\gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \mathbf{R}_k^{-1} \boldsymbol{\xi}_2) \quad (6.33)$$

This section is structured as follows:

1. A performance evaluation of the exact method is provided
  - for different image topologies,
  - in various combinations of blur and noise.
2. The influence of the inverse temperature parameter  $\beta$  on the estimation is analyzed. This influence is illustrated on a specific image topology that is rather sensitive to the choice of this parameter.
3. The estimation results of the exact method are compared to those of the two proposed approximations. This analysis is performed in order to evaluate the cost in terms of estimation quality that must be paid for the speed gain.

### 6.5.1 Evaluation of the exact method

Let us start our analysis with the evaluation of the exact method performances and a detailed interpretation of the results. Using the exact method means in fact that the samples of the complete textures  $\mathbf{x}_k$  are obtained through the sampling by Perturbation-Optimization method with a preconditioned gradient descent optimizer. Further on, we will also present segmentation results for different image topologies. This will allow us to evaluate the method versatility, adaptability and identify the potential weak spots.

Figure 6.4 illustrates the method's performance on a typical image topology, in an observation scenario with  $w_f = 4.5$  and  $\gamma_n = 60$ , corresponding to an  $\text{SNR} = 18\text{dB}$ .

This first example consists in an image composed of 4 regions and containing  $K = 3$  different classes of texture. In this case, the deconvolution-segmentation algorithm con-

verges to a label configuration very similar to the true labels, with only 1.2% of miss-classified pixels, despite the considerable degradation of the image. Moreover, the texture parameters relative estimation error is small, less than 5%. The full textures  $x_k$  are also accurately estimated, having the same characteristics as the original textures and also preserving the correct phase. The blur and the noise are eliminated and the resulting textured image strongly resembles the original image.

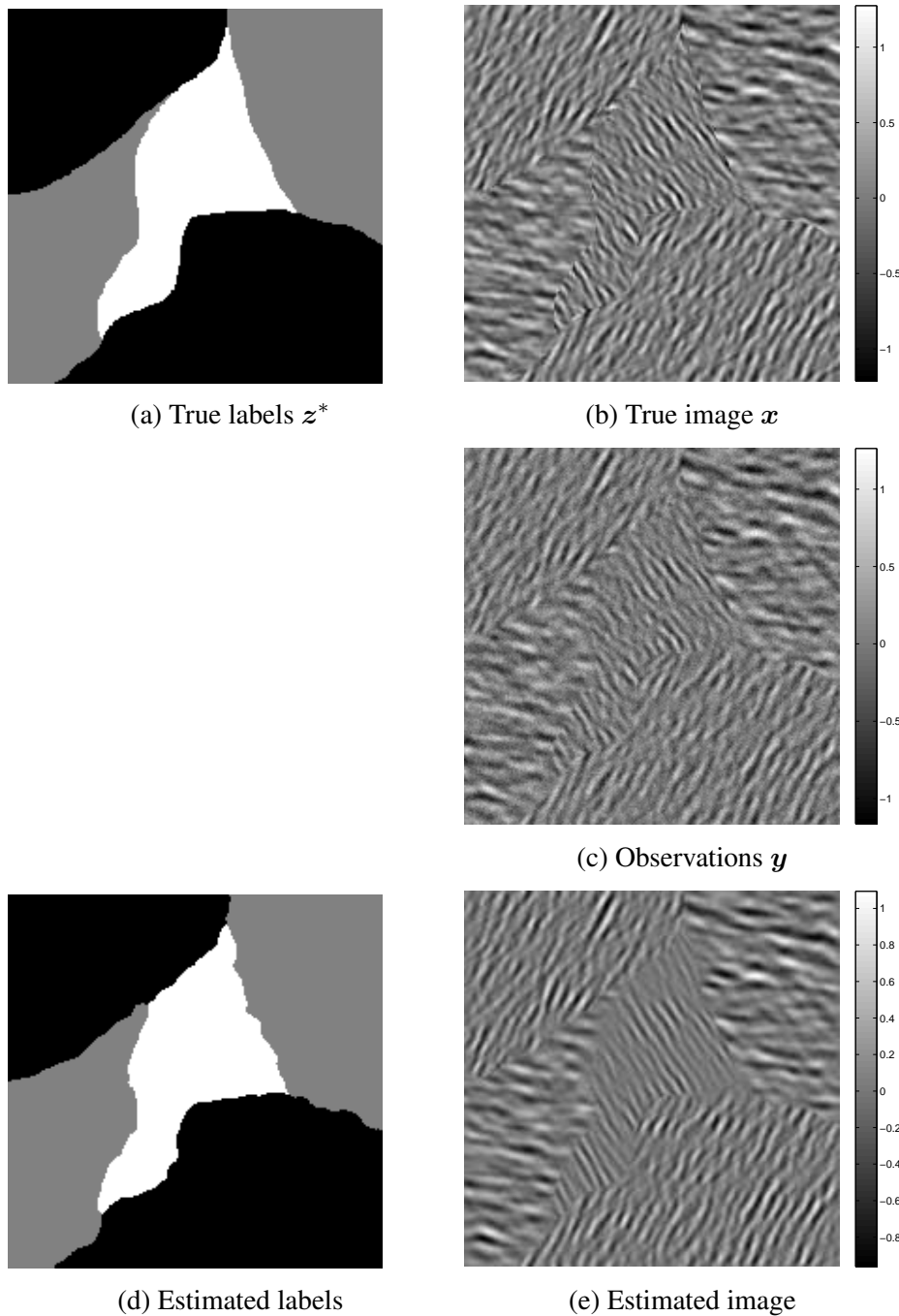


Figure 6.4: Segmentation results and reconstructed textured images. The temperature parameter is fixed to  $\beta = 1$ .

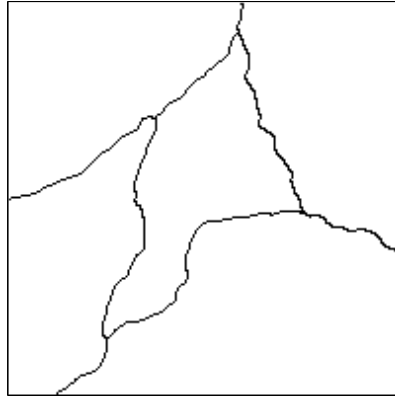


Figure 6.5: Contours

### 6.5.1.1 Label analysis

Our method is region-based, meaning that it provides closed contours, unlike a great part of the existing works in textured image segmentation. This can be seen in Figure 6.5 where we show the contour representing the set of boundary pixels between different regions, *i.e.*, the set of pixels for which at least one of its 4 neighbors has a different label.

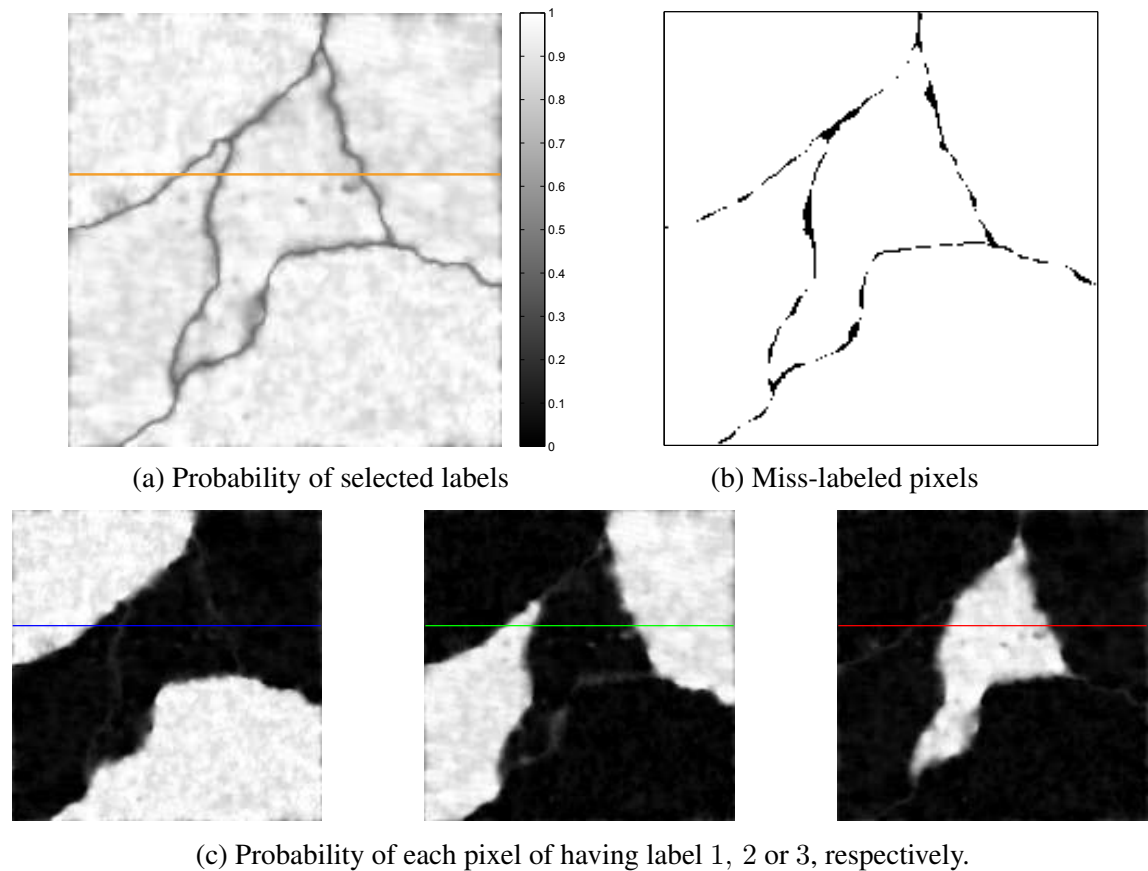


Figure 6.6: Link between the probability of the selected label and the estimation error. The probabilities of each possible label value.

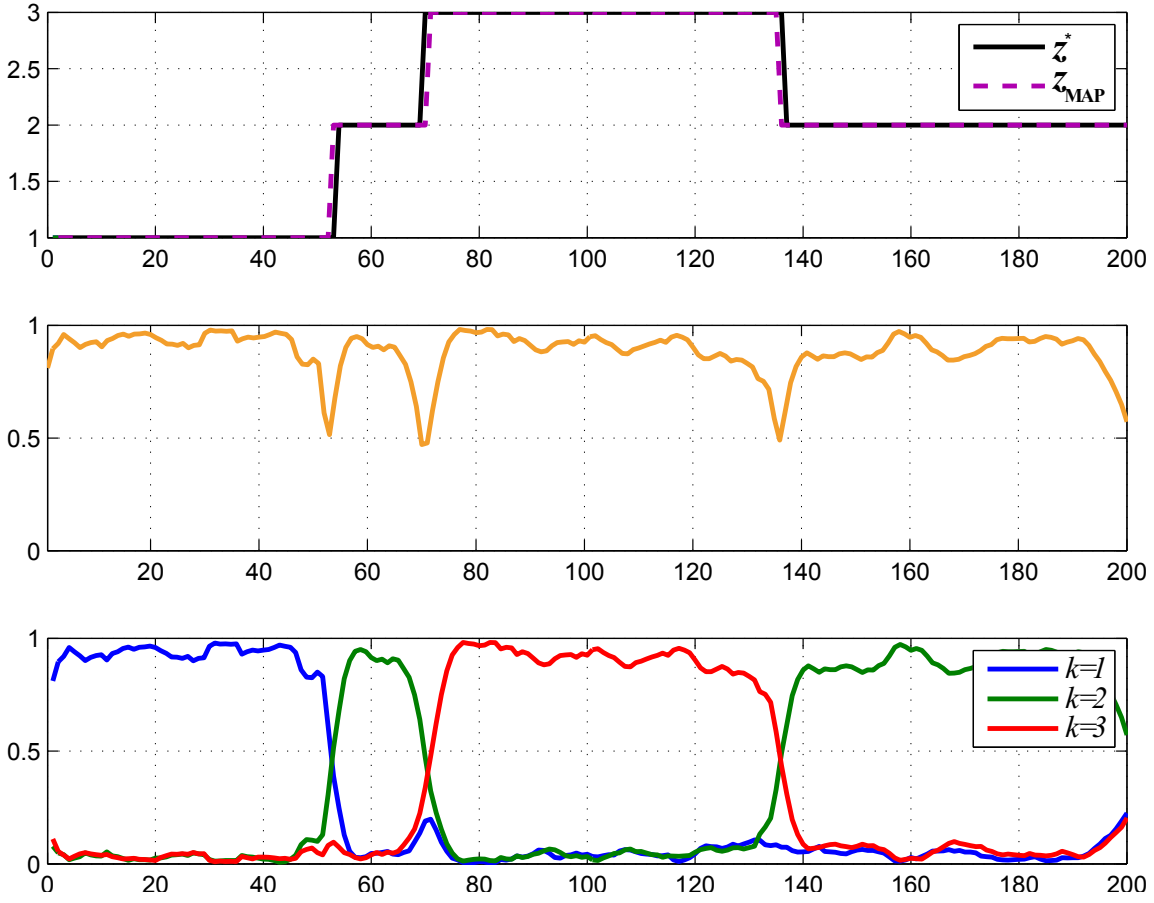


Figure 6.7: Cross-section of the true and estimated label fields, the probability of the selected labels and the probabilities of every possible label value  $k$ .

One of the main advantages of using a probabilistic approach is the fact that it not only provides estimates for the unknowns, but also information regarding the uncertainty associated to these estimates. Figures 6.6 and 6.7 illustrate our analysis on the label estimates and their probability, in a 2D form and as 1D cross sections, respectively.

Figure 6.6a shows the probabilities of the selected labels  $\hat{z} = z_{\text{MAP}}$ . As specified in the previous sections, the estimate is represented by the label having the maximum marginal posterior probability. Nevertheless, for certain pixels this maximum probability can have a small value, while for others a high value. Obviously, the estimate is more reliable in the second case. We can observe this fluctuation in Figure 6.6a, this probability field being obtained as the maximum of the three probability fields shown in Figure 6.6c.

The uncertainty associated to the chosen label is high at certain locations in the image and it is safe to assume that at these locations there is a higher chance of selecting a wrong label.

**Remark:** The colored lines in Figures 6.6a and 6.6c indicate the position of the 1D cross sections presented in Figure 6.7.

The analysis of the selected label posterior probability can be done on any image, even when the true label configuration is unknown. In order to verify if indeed we are more

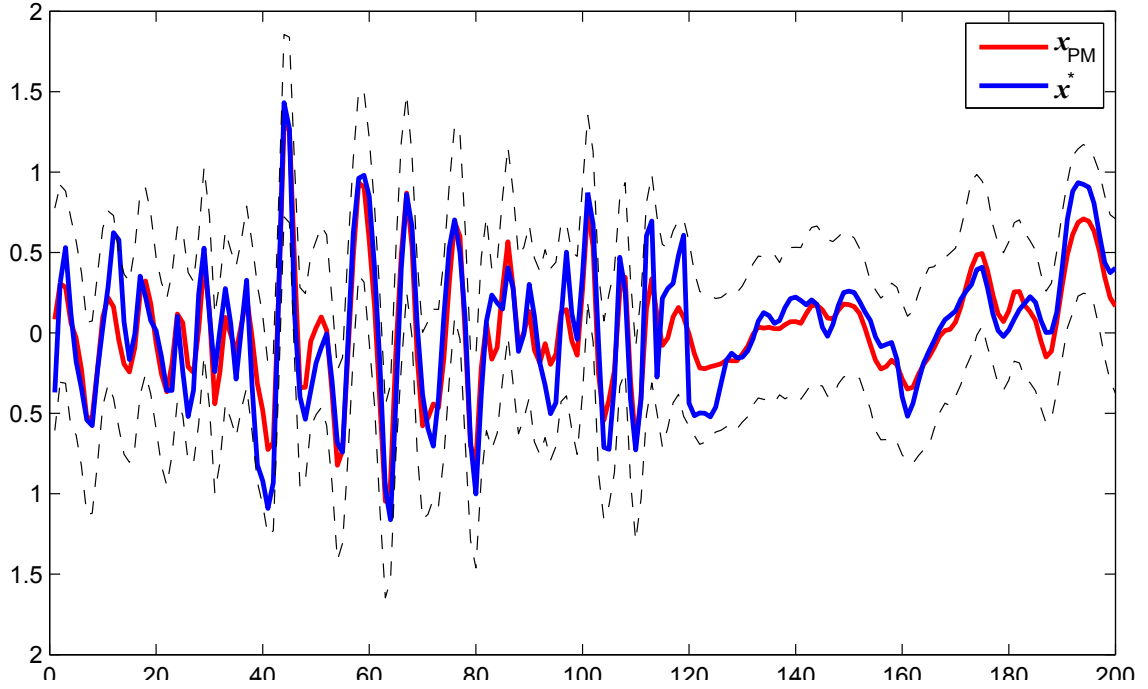


Figure 6.8: Cross-section of the true and estimated image. The dotted lines indicate the  $\pm 3\sigma$  interval around the PM estimate.

prone to error in the area with small posterior probability, we have compared our estimation result to the true labels in a case where they are known. We can immediately notice in Figure 6.6b that all of the miss-labeled pixels are in fact positioned in these areas of weaker posterior probability, shown in Figure 6.6a. This reinforces our statement concerning the utility of the probabilistic approach, due to its ability to identify and quantify the problem difficulty.

The first plot in Figure 6.7 shows the superposition of the true label field and the estimated labels. We can notice that the estimated labels are very close to the true labels, the only errors being recorded at the region edges, where the contours may be off by a few pixels. Nevertheless, there are no oscillations and the regions are correctly identified.

The second plot shows the probability of the aforementioned selected labels. For the majority of the selected labels, the probability is quite elevated, approaching 1. The regions where this probability drops superpose perfectly on the contours. This means that the less reliable estimations occur around the region boundaries.

The last plot illustrates the probabilities of each label, individually. The probabilities of the selected labels are simply the maxima of these  $k$  probabilities.

### 6.5.1.2 Restored image analysis

Figure 6.8 is dedicated to the analysis of the image restoration results and the posterior statistics of the estimate. This figure superposes a cross section of the original image and the estimated image. The first aspect that should be noticed is that the scale is correctly estimated. Also, overall, the textural characteristics are very accurately restored, *i.e.*, the

$\theta_k$  parameters estimates are accurate.

The dotted lines represent the  $-3\sigma$  and  $+3\sigma$  levels around the PM image estimate, where  $\sigma$  is the posterior standard deviation, computed empirically from the samples. This defines a confidence region around our estimate and we can notice that the true image is always confined inside this interval.

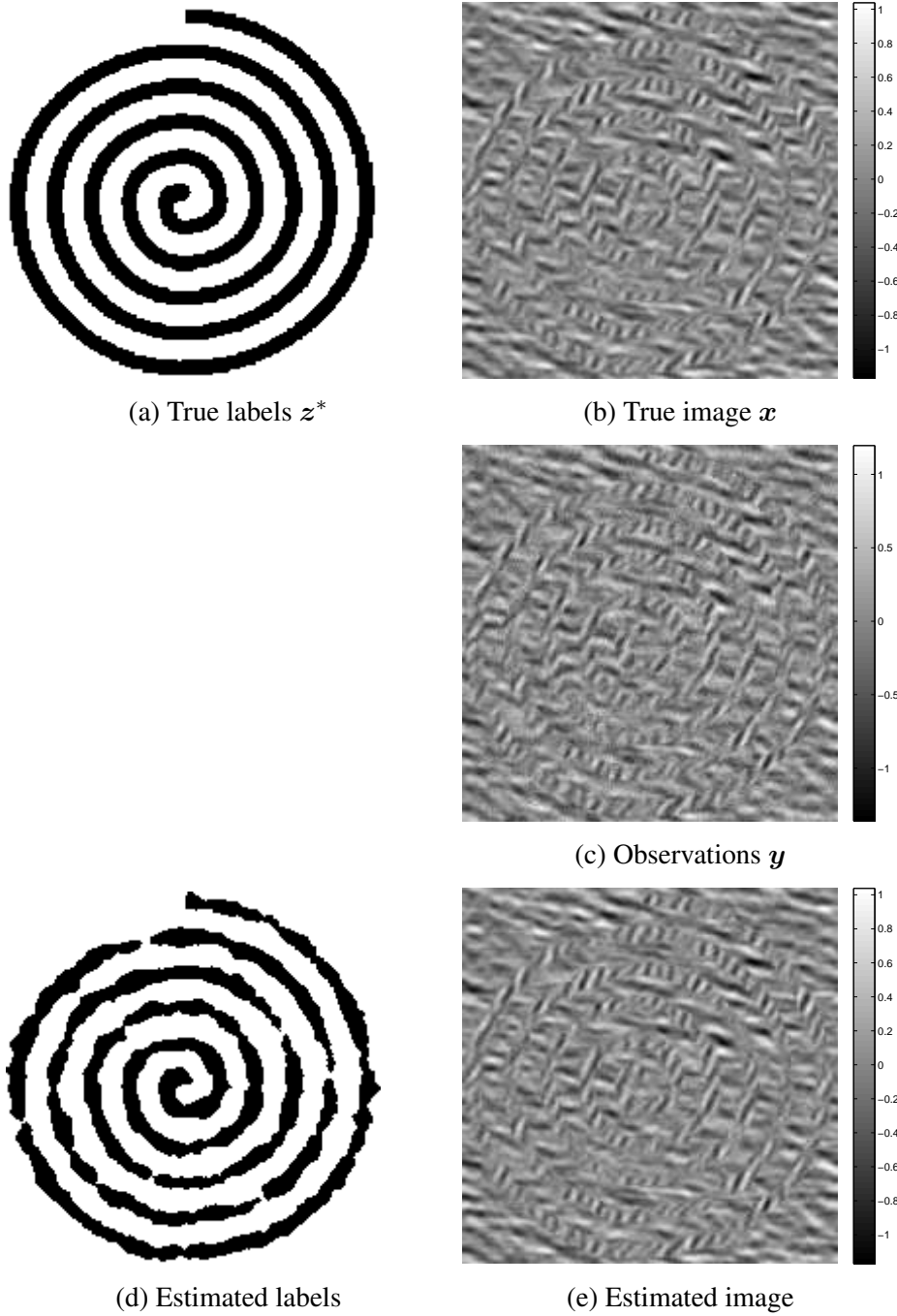


Figure 6.9: Segmentation results and reconstructed textured images. The temperature parameter is fixed to  $\beta = 0.7$ .



### 6.5.1.3 Other image topologies

In the case of the second image topology, illustrated in Figure 6.9, although the number of textures is more reduced  $K = 2$ , the task is more difficult due to the shape of the regions. The presence of a relatively thin, continuous structure makes the tuning of the  $\beta$  parameter more difficult. For this reason, the label estimation presents flaws, especially under the form of discontinuities in the original structure.

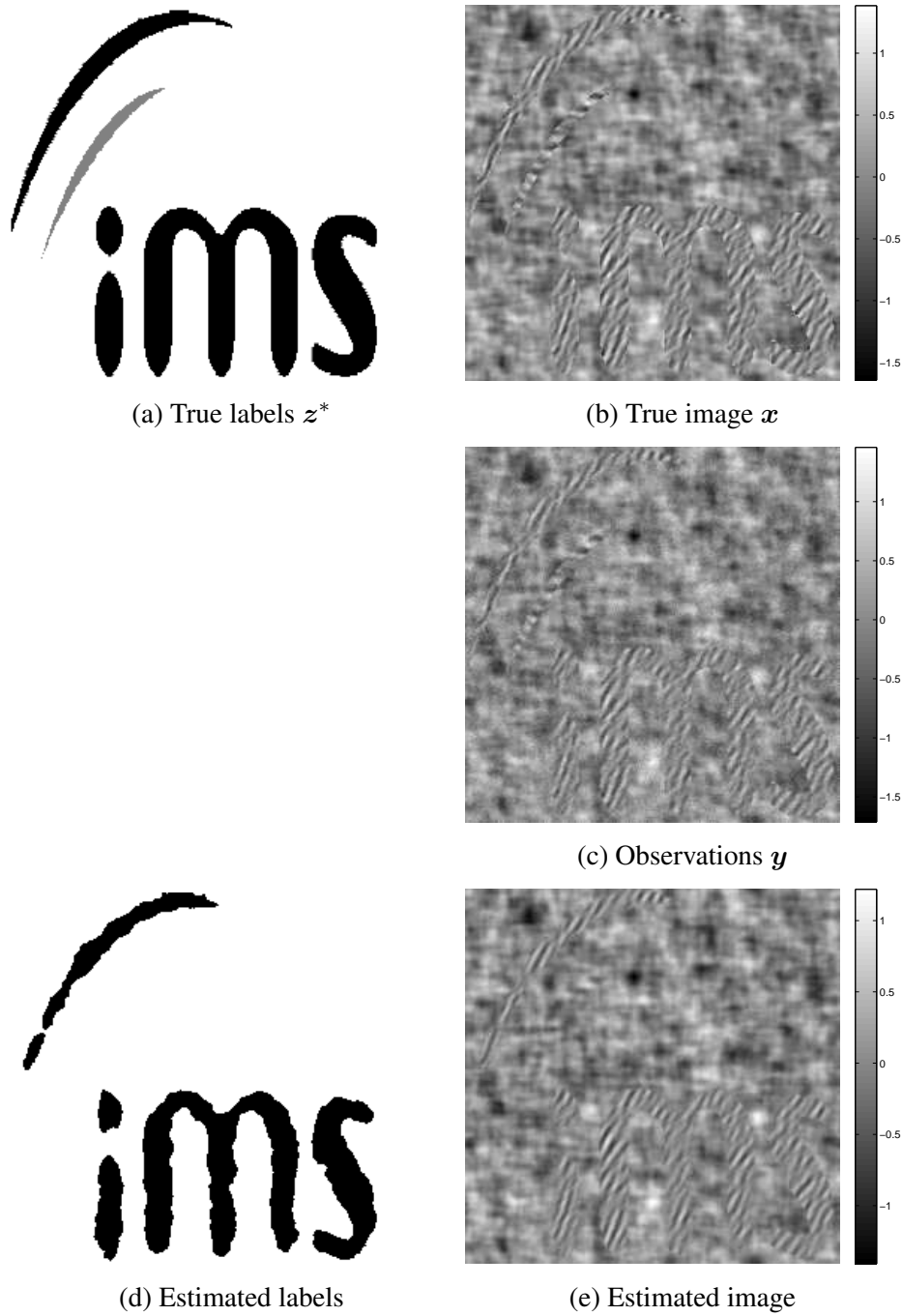


Figure 6.10: Segmentation results and reconstructed textured images. The temperature parameter is fixed to  $\beta = 0.9$ .



Figure 6.11: Segmentation results in an observation case  $w_f = 2$  and  $\gamma_n = 40$  corresponding to a blurred SNR = 21.5dB.

The third image, represented in Figure 6.10, permits the observation of another interesting aspect. Despite the good overall label estimation, there is a region that disappears completely. The particularity of this region is that although it has a significant size in terms of pixels, it has an elongated form and it is very thin. The reason behind this estimation error is again the  $\beta$  parameter and the representation capabilities of the Potts model itself, which seems to be more appropriate for representing compact, blobby regions rather than elongated structures. This statement is supported by the fact that the structure represented by the *point of the "i"* is correctly estimated, although it is smaller.

#### 6.5.1.4 Blur and noise influence

Further results show how the estimation is affected by various observation conditions. Figure 6.11 illustrates the method performances in a weaker convolution case  $w_f = 2$  and higher level of noise  $\gamma_n = 40$ .

The method performs very well in this case, the estimated label field being very close to the true labels. The superior quality of the estimates as compared to the results in Figure 6.9 can be explained by the fact that the observations are less degraded (an SNR of 21dB as opposed to 19dB) despite the higher noise level.

#### 6.5.2 Influence of the $\beta$ parameter

Figure 6.12 illustrates the influence of the  $\beta$  parameter in the case where we have encountered the aforementioned difficulty regarding the elongated structure. This test is performed in a case with no convolution. It is obvious that the choice of  $\beta$  significantly affects the estimation results. A small value permits to identify all the regions, even the thin ones, however, at the cost of producing parasitic regions. For a too large value, there are no parasitic regions, nevertheless, the thin structures are lost. In this work, the value of  $\beta$  is set by manual tuning. In cases where  $\beta$  must be tuned automatically, it can be included in the estimation process, however, the results will probably be less accurate than in the manual tuning case.

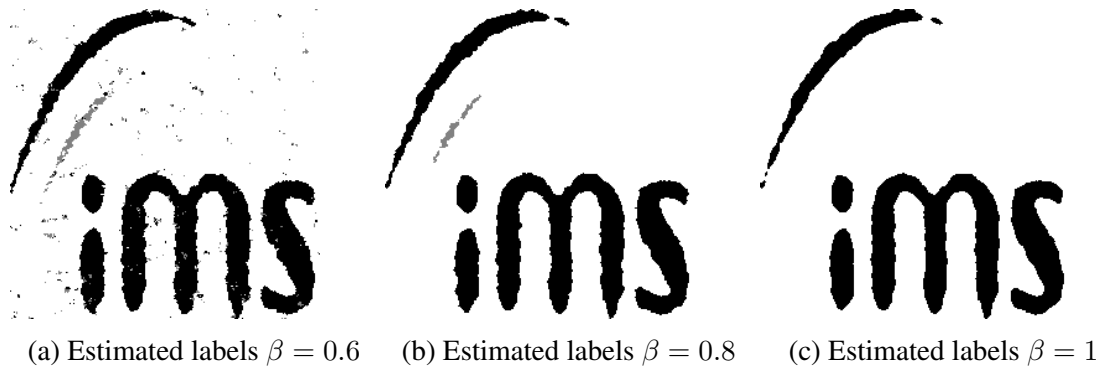


Figure 6.12: Segmentation results and reconstructed textured images for the two approximations in an observation case  $w_f = 0$  and  $\gamma_n = 50$  corresponding to an  $\text{SNR} = 24\text{dB}$ .

### 6.5.3 Influence of the approximations

Figure 6.13 shows the estimation results in a case with no blur. Thus, this is a particular situation of denoising-segmentation. The "exact method", consisting in sampling the  $x_k$  textures using the sampling by Perturbation-Optimization algorithm is compared to the two approximations that we have proposed in Section 6.4.2.2 in order to enhance the speed performances. In this case, the two approximations for sampling the textures by FFT are equivalent, since there is no blur. This is in fact confirmed by the estimation results,

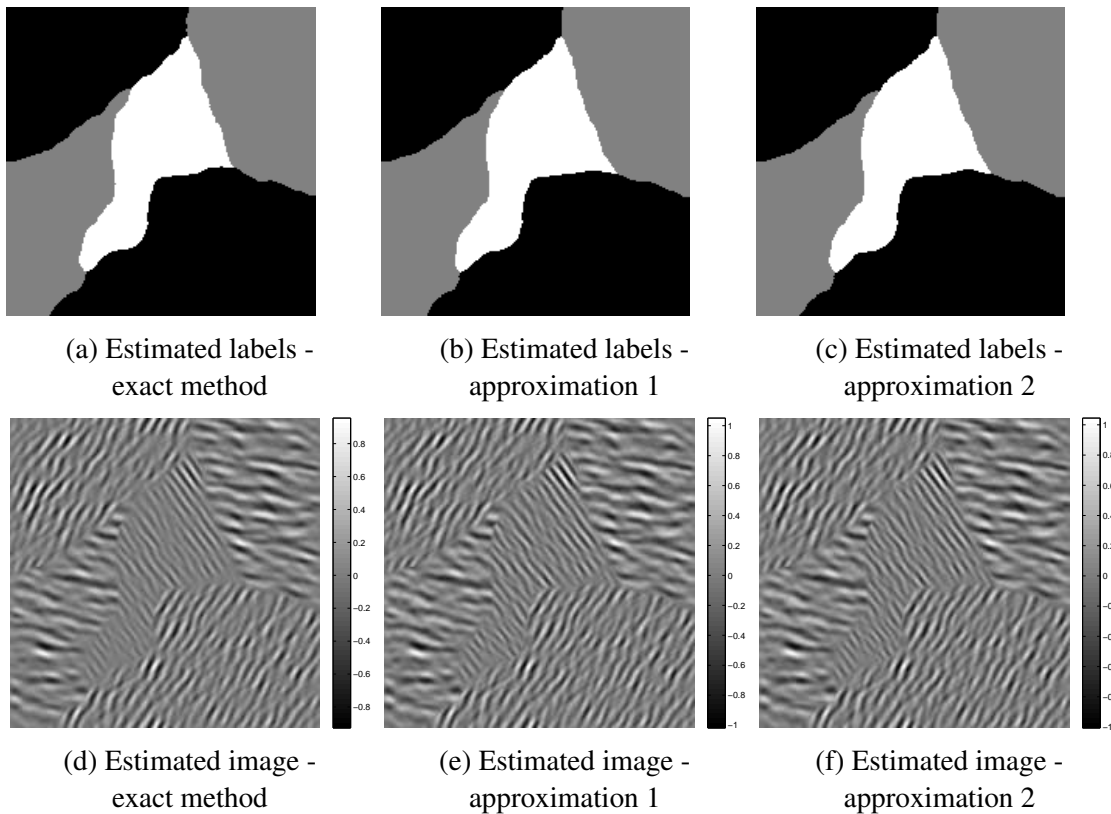


Figure 6.13: Segmentation results and reconstructed textured images for the two approximations in an observation case  $w_f = 0$  and  $\gamma_n = 10$  corresponding to an  $\text{SNR} = 8\text{dB}$ .

which are very similar for the two approximations. Moreover, when compared to the exact sampling method, the results provided by the approximations based methods are practically indistinguishable. This means that the speed gain provided by the use of the approximations does not come at the cost of poor quality estimation for the labels and the image.

## 6.6 Conclusion and Perspectives

This chapter has presented our method for joint image deconvolution and segmentation, dedicated to textured images. This is a very difficult problem due to the large amount of unknowns and the complicated dependencies of the variables. The formulation of the problem itself has not been a trivial task and has demanded a long reflection time in order to find the best manner to accurately express the hierarchical dependencies. In this context, the most adapted choice was to model  $K$  full images, corresponding to each class, rather than model the pixels of the  $x$  image. This has allowed us to obtain an expression of the posterior in which the dependency with respect to each unknown has a relatively convenient form.

Our Bayesian method relies on sampling the posterior conditional laws. Based on these samples, estimators for the labels and the image are built. Nevertheless, the sampling process for the full textures and for the labels has also proved to be rather challenging and has demanded a certain amount of ingenuity to overcome the impasses. The proposed methodological and algorithmic original aspects have contributed to developing a method that is both theoretically sound and efficient for solving this difficult problem never tackled before.

The previous section has presented the results of a series of tests performed on various convolution and noise conditions, for different image topologies and for several values of the Potts field temperature parameter. These results have shown that the method is able to accurately segment the image, provide a good estimation for the textures parameters and thus restore the original image (affected by noise and blur).

Our tests have shown however that the method is sensitive to the tuning of the temperature parameter  $\beta$ . This leads us to the future developments of this work:

- the first perspective of our current work is to automatically estimate  $\beta$ ;
- a second future contribution is the use of our SMGRF model for the constituent textures  $x_k$ . Nevertheless, this will add an extra layer of sampling for the auxiliary variables. Except for the extra computational time, this extension should be straightforward and should not pose any methodological difficulty;
- the third future development of this work aims at performing a myopic deconvolution, *i.e.*, considering that  $w_f$ , the width of the convolution filter, is unknown and estimating it along with the rest of the parameters.

As it can be seen from this brief listing of the perspectives, the work on this topic is far from being over. Nevertheless, even in its current form, the method presented in

this chapter addresses a problem that had not been tackled so far, while achieving very satisfactory results.



# Conclusion and Perspectives

---

This work has presented a wide range of inverse problems of central importance in image processing, translated in the case of textured images. In this context, a series of advances have been made on an algorithmic, modeling and methodological level.

The **algorithmic contribution** consists in **developing an efficient version of the Metropolis-Hastings algorithm**. The proposition law of this new sampler is based on a component resembling the Newton direction, in which the inverse Hessian is replaced by the Fisher matrix. As a result, the problems related to taking the inverse of the Hessian matrix are avoided and the directional component of the proposal is always sure to have the direction of gradient descent. Moreover, in the case of our particular *a posteriori* conditional law, this proposal can be expressed only based on the gradient of the law.

As for the perspectives of this work, this sampler can be integrated in a wide spectrum of applications in order to accelerate the sampling process.

\*\*\*\*\*

The **model-related contribution** consists in **developing a model for texture analysis and synthesis based on a Scale Mixture of Gaussian Random Fields with parametric Power Spectral Density**. The parametric model for the Power Spectral Density and the values of its parameters encode the textural characteristics and allow us to classify the texture or to generate new texture samples having the same features as the original one. This non-Gaussian model is built using a set of auxiliary variables such that the law of the image, conditionally on the auxiliary variables, is Gaussian and, marginally with respect to these variables, it is no longer Gaussian. In the current work, for algorithm efficiency purposes, these auxiliary variables are considered independent.

The perspectives regarding this direction of research are to use correlated priors for the auxiliary variables. This means that the auxiliary variables will no longer be independent. This would probably trigger an increase of the representation capabilities of our texture model. Nevertheless, the side effect is an increased computational load for sampling a texture. Another possible development is the use of more complex shapes for the Power Spectral Density and investigating how the phase of the Fourier Transform coefficients can be used to obtain more complex models.

\*\*\*\*\*

The **methodological contribution** is threefold, since we have addressed separately three inverse problems in the context of indirect observations of textured images.

Firstly, we have devised a method for **myopic deconvolution and parameter estimation for textured images corrupted by a blur and noise**. These textured images are modeled by a Gaussian Random Field with parametric Power Spectral Density. The Point Spread Function also has a parametric exponential form. The aforementioned parameters are unknown and are estimated by the method.

This work can be immediately extended by using more complicated form for the convolution filter. Another extension can be the use of the Scale Mixture of Gaussians model for the textured image.

\*\*\*

The second methodological contribution is an **evidence-based classification method for indirectly observed textured images**. As in the previous case, the textured images are affected by a blur and by noise and the goal is to determine the model and the parameters of the texture's Power Spectral Density. In this case, the convolution filter is known, thus we are not in a myopic deconvolution case, however, the noise level is unknown and must be estimated. The textures are modeled either by Scale Mixture of Gaussian Random Fields or by Gaussian Random Fields and the method selects the most probable model to have generated the realization. The method relies on sampling and the algorithmic implementation embeds the efficient Fisher matrix based sampler.

This work can be continued by extending the dictionary of shapes for the Power Spectral Density. Another future development is the estimation of the convolution filter, along with the rest of the unknowns.

\*\*\*

The final methodological advance is our **deconvolution-segmentation method for textured images**. As in the classification case, the convolution filter is known and the noise level is estimated. This method achieves very satisfactory results in the case of a very complicated problem. Texture segmentation in itself is a very challenging task, whilst the literature holds no accounts of joint deconvolution-segmentation methods for textured images. Our method is based on a Potts model for the labels, with a temperature parameter that is manually tuned, and on a Gaussian Random Field model for the textures, with a given set of models for the Power Spectral Density, but unknown parameters.

Besides the significant theoretical challenges related to this problem, the implementation of this method has confronted us with the difficult task of sampling the labels and the pixels in an efficient manner, to avoid having a prohibitive computational load. This constraint has gradually guided us towards the final modeling of this problem that is presented in this manuscript. Nevertheless, this is the result of a series alternatives that we have explored. Among all these formulations, the current version is the only one that allowed us to accurately represent the variable dependencies and to obtain good speed performances.

This topic can be further developed by integrating an automatic estimation of the temperature parameter, in order to permit the method to adapt to any image topology without user intervention. Another extension can be to also estimate the convolution filter.

\*\*\*

Finally, an ambitious project would be to combine all these problems in order to obtain



a method that, starting from blurred and noisy observations of an image made up of several regions, each with its own texture, would be able to estimate the convolution filter, segment the image and determine the model of the Power Spectral Density and the values of its parameters for each of the constituent textures.

\*\*\*\*\*

All the methods we have devised are based on optimal estimators, such as the Posterior Mean for the parameter estimation and the evidence-based classifier for the model selection. Consequently, the methods themselves are optimal from the Mean Square Error and the Mean Classification Risk point of view.

The goal of this work was to provide answers to questions that have not yet been posed or answered in the literature in the context of indirect observations of textured images.



# Fisher information for indirectly observed GRF textures

---

The data may carry different amounts of information, this affecting directly the estimation performances. The amount of available information depends on the SNR =  $\gamma_n/\gamma_x$  or the particular parameter values. The mean available information on  $\Psi$  is quantified by the Fisher information matrix. We focus on each parameter  $\psi$  (any one of  $\Psi$ 's elements), through its diagonal elements: the expectation of the log-likelihood's second derivative:

$$\mathcal{J}(\psi) = -\mathbb{E}_{\mathbf{y}|\psi} \left[ \frac{\partial^2}{\partial \psi^2} \log f(\mathbf{y}|\Psi) \right] \quad (\text{A.1})$$

$f(\mathbf{y}|\Psi)$  can be obtained from (2.4) and the texture models:  $\mathbf{y}|\Psi \sim \mathcal{N}(0, \mathbf{R}_y(\Psi))$ , with  $\mathbf{R}_y(\Psi) = \mathbf{H}_\eta^t \mathbf{R}_x(\boldsymbol{\theta}) \mathbf{H}_\eta + \mathbf{R}_n(\gamma_n)$ . The variance of  $\dot{y}_p|\Psi$  becomes:

$$r_p(\Psi) = \frac{1}{\gamma_x} \frac{g_p(\eta)}{\lambda_p(\boldsymbol{\theta})} + \frac{1}{\gamma_n} \quad (\text{A.2})$$

where  $g_p(\eta) = |\dot{h}_p(\eta)|^2$ .

After derivation, the expression contains first and second order derivatives of  $r_p(\Psi)$  with respect to  $\psi$ . When taking the expectation, knowing that  $\mathbb{E}_{\mathbf{y}|\psi} [|\dot{y}_p|^2] = r_p(\Psi)$ , the second order derivatives cancel out. Thus:

$$\mathcal{J}(\psi) = \sum_{p=1}^P \left[ \frac{1}{r_p(\Psi)} \cdot \frac{\partial}{\partial \psi} r_p(\Psi) \right]^2 \quad (\text{A.3})$$

where  $r'_p$  is the derivative of  $r_p$  w.r.t. the current parameter  $\psi$ . Then:

$$\mathcal{J}(\gamma_n) = \gamma_n^{-2} \sum_p \left[ 1 + \gamma_n/\gamma_x \cdot \frac{g_p(\eta)}{\lambda_p(\boldsymbol{\theta})} \right]^{-2} \quad (\text{A.4})$$

$\mathcal{J}(\gamma_n)$  is a decreasing function, hence, the smaller the  $\gamma_n$  (the higher the noise level), the easier it is to estimate its level.

Similarly,

$$\mathcal{J}(\gamma_x) = \gamma_x^{-2} \sum_p \left[ 1 + \gamma_x/\gamma_n \cdot \frac{\lambda_p(\boldsymbol{\theta})}{g_p(\eta)} \right]^{-2} \quad (\text{A.5})$$

is also a decreasing function, *i.e.*, the smaller the  $\gamma_x$  (higher signal level), the easier its estimation.

For the texture parameters  $\theta \in \boldsymbol{\theta}$ :

$$\mathcal{J}(\theta) = \sum_p \left\{ \frac{g_p(\eta) \cdot \lambda'_p(\boldsymbol{\theta})}{\lambda_p(\boldsymbol{\theta}) [\gamma_x/\gamma_n \cdot \lambda_p(\boldsymbol{\theta}) + g_p(\eta)]} \right\}^2 \quad (\text{A.6})$$

Another interesting case is the noiseless scenario ( $\gamma_n = \infty$ ):

$$\mathcal{J}(\theta) = \sum_{p=1}^P \left[ \frac{1}{\lambda_p(\boldsymbol{\theta})} \cdot \lambda'_p(\boldsymbol{\theta}) \right]^2 \quad (\text{A.7})$$

The Fisher information regarding  $\theta$  depends only on the PSD and its parameters (as in [VGB11]). This means that the amount of information for the texture parameters does not depend on the form of the TF.

The same considerations hold for the TF parameter  $\eta$ :

$$\mathcal{J}(\eta) = \sum_p \left[ \frac{g'_p(\eta)}{\gamma_x/\gamma_n \cdot \lambda_p(\boldsymbol{\theta}) + g_p(\eta)} \right]^2 \quad (\text{A.8})$$

thus, for low SNR,  $\mathcal{J}(\eta)$  is small and for no noise,  $\mathcal{J}(\eta)$  only depends on the TF:

$$\mathcal{J}(\eta) = \sum_p \left[ \frac{1}{g_p(\eta)} \cdot g'_p(\eta) \right]^2.$$

# Optimal Bayesian Estimation

---

A Bayes estimator is the optimal estimator with respect to a certain cost function. This estimator minimizes the expected value with respect to the parameter and the data of the cost function, also known as Bayesian risk.

The estimators employed in this work, the PM and the evidence based classifier, are optimal from the Mean Squared Error (MSE) and the Mean Classification Error (MCE) standpoint, respectively.

## B.1 Posterior Mean

Let us consider the  $\mathbf{y}$  be the observations and  $\theta$  be the unknown parameter, with prior distribution  $\pi(\theta)$ .

Based on the observations, the estimator  $\delta(\mathbf{y}) = \bar{\theta}(\mathbf{y})$  can be built. A cost function  $C(\theta, \bar{\theta})$  will quantify the significance of being wrong. An optimal Bayesian estimator minimizes the Bayesian risk function, *i.e.*, the posterior expected cost:

$$\begin{aligned} \rho(\pi(\theta), \bar{\theta}(\mathbf{y})|\mathbf{y}) &= \mathbb{E}_{\theta|\mathbf{y}} [C(\theta, \bar{\theta}(\mathbf{y}))] \\ &= \int_{\theta} C(\theta, \bar{\theta}(\mathbf{y})) \cdot f(\theta|\mathbf{y}) d\theta \end{aligned} \quad (\text{B.1})$$

By averaging the cost function over all the possible observations and all the possible values of the unknown, we obtain the Bayes risk:

$$\begin{aligned} R(\pi(\theta), \bar{\theta}(\cdot)) &= \int_{\theta} \int_{\mathbf{y}} C(\theta, \bar{\theta}(\mathbf{y})) \cdot f(\mathbf{y}|\theta) \cdot \pi(\theta) d\mathbf{y} d\theta \\ &= \int_{\theta} \int_{\mathbf{y}} C(\theta, \bar{\theta}(\mathbf{y})) \cdot f(\mathbf{y}, \theta) d\mathbf{y} d\theta \end{aligned} \quad (\text{B.2})$$

Then, the decision function that minimizes this risk is:

$$\begin{aligned} \min_{\bar{\theta}(\mathbf{y})} R(\pi(\theta), \bar{\theta}(\cdot)) &= \min_{\delta(\mathbf{y})} \int_{\theta} \int_{\mathbf{y}} C(\theta, \bar{\theta}(\mathbf{y})) \cdot f(\mathbf{y}, \theta) d\mathbf{y} d\theta \\ &= \min_{\bar{\theta}(\mathbf{y})} \int_{\mathbf{y}} \left[ \int_{\theta} C(\theta, \bar{\theta}(\mathbf{y})) \cdot f(\theta|\mathbf{y}) d\theta \right] f(\mathbf{y}) d\mathbf{y} \end{aligned}$$

$$\begin{aligned}
&= \min_{\bar{\theta}(\mathbf{y})} \int_{\mathbf{y}} \rho(\pi(\theta), \bar{\theta}(\mathbf{y})|\mathbf{y}) f(\mathbf{y}) d\mathbf{y} \\
&= \int_{\mathbf{y}} \left[ \min_{\bar{\theta}(\mathbf{y})} \rho(\pi(\theta), \bar{\theta}(\mathbf{y})|\mathbf{y}) \right] f(\mathbf{y}) d\mathbf{y}
\end{aligned}$$

The last equality holds true because the cost function is non-negative by definition, thus  $\rho(\pi(\theta), \bar{\theta}(\mathbf{y})|\mathbf{y}) \geq 0$  and minimizing the integral of a non-negative function is equivalent to minimizing the function at each point.

Let us now consider the cost function as being the squared error:

$$C(\theta, \bar{\theta}) = (\theta - \bar{\theta})^2 \quad (\text{B.3})$$

We can then determine the estimator that minimizes the MSE by computing:

$$\min_{\bar{\theta}(\mathbf{y})} \int_{\theta} (\theta - \bar{\theta}(\mathbf{y}))^2 \cdot f(\theta|\mathbf{y}) d\theta \quad (\text{B.4})$$

which results in:

$$\bar{\theta}(\mathbf{y}) = \int_{\theta} \theta \cdot f(\theta|\mathbf{y}) d\theta \quad (\text{B.5})$$

Consequently, the estimator that minimizes the MSE is the PM.

## B.2 Evidence based Classifier

Let us now consider that the data  $\mathbf{y}$  are realizations of the law  $f(\mathbf{y}|\mathcal{M}^*)$ . The goal is to determine the value of  $\mathcal{M}$  which is the most likely, from the Bayes risk point of view, to have generated the data. In this case,  $\mathcal{M}$  is a discrete variable, with prior probability  $\pi(\mathcal{M})$ .

Similarly, a decision function  $\Delta(\mathbf{y}) = \bar{\mathcal{M}}(\mathbf{y})$  will select a certain model, based on the current observations. The cost function  $C(\mathcal{M}^*, \bar{\mathcal{M}}(\mathbf{y}))$  quantifies the cost of choosing the wrong model. The Bayes risk writes in this case:

$$R(\pi(\mathcal{M}), \bar{\mathcal{M}}(\cdot)) = \sum_{\mathcal{M}} \int_{\mathbf{y}} C(\mathcal{M}, \bar{\mathcal{M}}(\mathbf{y})) \cdot f(\mathbf{y}, \mathcal{M}) d\mathbf{y} \quad (\text{B.6})$$

The optimal decision function, *i.e.*, the decision that minimizes the Bayes risk is:

$$\Delta(\mathbf{y}) = \arg \min_{\mathcal{M}} \sum_{\mathcal{M}} C(\mathcal{M}, \hat{\mathcal{M}}) \cdot f(\mathcal{M}|\mathbf{y}) \quad (\text{B.7})$$

One of the most commonly used cost functions in classification is the 0/1 cost

$C(\mathcal{M}, \mathcal{M}^*) = 1 - \delta(\mathcal{M}, \mathcal{M}^*)$ . Then,

$$\begin{aligned}
 \Delta(\mathbf{y}) &= \arg \min_{\hat{\mathcal{M}}} \left\{ \left[ \sum_{\mathcal{M}} f(\mathcal{M}|\mathbf{y}) \right] - f(\hat{\mathcal{M}}|\mathbf{y}) \right\} \\
 &= \arg \min_{\mathcal{M}} [1 - f(\mathcal{M}|\mathbf{y})] \\
 &= \arg \max_{\mathcal{M}} [f(\mathcal{M}|\mathbf{y})] \\
 &= \Delta_{\text{MAP}}(\mathbf{y})
 \end{aligned} \tag{B.8}$$





## APPENDIX C

# Potts Model

---

The Potts model originates from statistical mechanics and was used to describe the particle interaction in a crystalline lattice. The model has later been introduced to image processing and more specifically to image segmentation, since it is appropriate for modeling the prior behavior of the labels.

In a Potts model with  $K$  states, each node of a graph is assigned an integer value between 1 and  $K$ , the assembly of all these assignments being called a configuration. The set of all possible configurations is  $\{1 \dots K\}^P$ , where  $P$  is the total number of pixels.

The energy of a configuration  $\mathbf{z}$  is denoted as  $E(\mathbf{z})$ , representing the number of edges in the graph with endpoints labeled differently. It has the form:

$$E(\mathbf{z}) = \sum_{s \sim t} \delta(z_s, z_t) \quad (\text{C.1})$$

= number of alike neighboring pairs,

where  $\delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{if } a \neq b \end{cases}$ .

A probability distribution function, known as the Gibbs distribution is assigned to the set of states:

$$g(\mathbf{z}) = \mathcal{Z}_z(\beta)^{-1} \cdot \exp[\beta \cdot E(\mathbf{z})] \quad (\text{C.2})$$

where  $\mathcal{Z}_z(\beta)$  is the normalizing constant that makes  $g$  a probability distribution. Often, this quantity is also referred to as the partition function. The parameter  $\beta$  is a measure of the inverse temperature. The value of  $\beta$  influences the structure of the resulting Potts field as follows:

- for low  $\beta$  the Potts field will have small regions and an overall noise-like character,

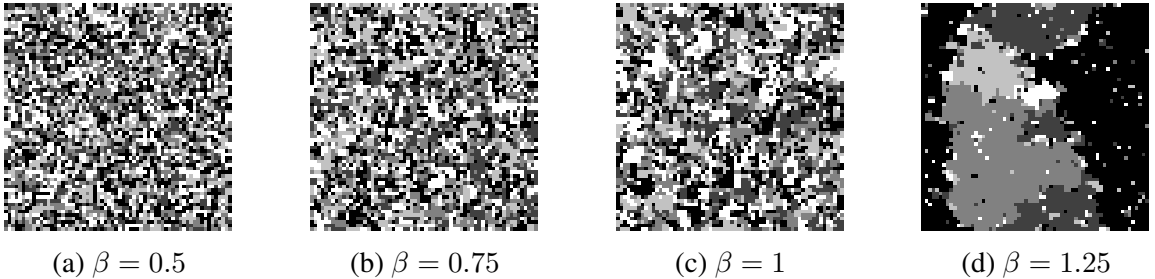


Figure C.1: Potts field realizations of size  $64 \times 64$  for  $S = 5$  states and various values of the temperature parameter.

- for high  $\beta$  the Potts field will be composed of big homogeneous regions, as the labels will have an increased tendency to aggregate.

The partition function has the expression:

$$\mathcal{C}_z(\beta) = \sum_{z \in \{1 \dots K\}^P} \exp \left[ \beta \sum_{s \sim t} \delta(z_s, z_t) \right] \quad (\text{C.3})$$

and is an intractable quantity, except for the special case  $K = 2$ , the Ising model [Gio08, Gio11]. This intractability is due to the number of terms in the sum that increases exponentially with  $P$ .

# Truncation Matrices

The extraction matrices  $\mathbf{S}_k$  from Chapter 6 perform at the same time two tasks:

- they extract the pixels having label  $k$ ,
- they perform zero-padding, *i.e.*, they replace the pixels having labels  $l \neq k$  with zero.

This can be formalized by passing through the intermediary matrices  $\mathbf{T}_k$ , for which:

$$\mathbf{S}_k = \mathbf{T}_k^t \mathbf{T}_k \quad (\text{D.1})$$

The  $\mathbf{T}_k$  matrices are binary truncation matrices, based on the labels  $\mathbf{z}$ . They are used to extract the pixels with label  $k$  from the full image  $\mathbf{x}_k$ . Let us consider that:

$$\begin{array}{ccc} \dots & & \\ \mathcal{J}_k = \{i | \mathbf{z}(i) = k\} & \rightarrow \mathbf{T}_k : & \underbrace{\mathbf{x}_{\mathcal{J}_k}}_{\text{card } \mathcal{J}_k \times 1} = \underbrace{\mathbf{T}_k}_{\text{card } \mathcal{J}_k \times P} \cdot \underbrace{\mathbf{x}_k}_{P \times 1} \\ \dots & & \end{array}$$

Then the  $\mathbf{T}_k$  matrices are of size  $\text{card } \mathcal{J}_k \times P$  and have the structure:

$$\mathbf{T}_k = \left\{ \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}}_P \right\}_{\text{card } \mathcal{J}_k}$$

The special properties of these matrices are that they cover the entire pixel lattice and that they select each pixel only once:

$$\mathbf{T}_1^t \mathbf{T}_1 + \mathbf{T}_2^t \mathbf{T}_2 + \dots + \mathbf{T}_K^t \mathbf{T}_K = \mathbf{I}_P \quad (\text{D.2})$$

$$\mathbf{T}_k \mathbf{T}_k^t = \mathbf{I}_{\text{card } \mathcal{J}_k} \quad (\text{D.3})$$

The second step of the process is the zero-padding, *i.e.*, arranging the selected pixels on their original positions in the image and replacing of the pixels corresponding to labels  $l \neq k$  by zeros. This can be achieved using the  $\mathbf{T}_k$  matrix as follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{T}_k^t \mathbf{x}_{\mathcal{J}_k} \\ &= \mathbf{T}_k^t \mathbf{T}_k \mathbf{x}_k \end{aligned} \quad (\text{D.4})$$



# Bibliography

- [ABDF11] M.V. Afonso, J.M. Bioucas-Dias, and M. Figueiredo, *An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems*, IEEE Transactions on Image Processing **20** (2011), no. 3, 681–695. [↑40](#)
- [ACS05] B. Abraham, O.I. Camps, and M. Sznai, *Dynamic texture with Fourier descriptors*, Proceedings of the 4th International Workshop on Texture Analysis and Synthesis, 2005, pp. 53–58. [↑23](#)
- [AG03] S. Arivazhagan and L. Ganesan, *Texture segmentation using wavelet transform*, Pattern Recognition Letters **24** (2003), 3197–3203. [↑22](#), [74](#)
- [AGBB12] S. Alpert, M. Galun, A. Brandt, and R. Basri, *Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration*, IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012), no. 2, 315–327. [↑74](#)
- [Aka74] H. Akaike, *A new look at the statistical model identification*, IEEE Transactions on Automatic Control **19** (1974), no. 6, 716–723. [↑54](#)
- [AKJ07] U.A. Ahmad, K. Kidiyo, and R. Joseph, *Texture features based on Fourier transform and Gabor filters: an empirical comparison*, Proceedings of the International Conference on Machine Vision, 2007, pp. 67–72. [↑23](#)
- [AMD10] H. Ayasso and A. Mohammad-Djafari, *Joint NDT Image Restoration and Segmentation Using Gauss-Markov Potts Prior Models and Variational Bayesian Computation*, IEEE Transactions on Image Processing **19** (2010), no. 9, 2265–2277. [↑75](#)
- [Ayk98] R.G. Aykroyd, *Bayesian Estimation for Homogeneous and Inhomogeneous Gaussian Random Fields*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998), no. 5, 533–539. [↑32](#)
- [BDF10] J.M. Bioucas-Dias and M. Figueiredo, *Multiplicative Noise Removal Using Variable Splitting and Constrained Optimization*, IEEE Transactions on Image Processing **19** (2010), no. 7, 1720–1730. [↑40](#)
- [Bea03] M.J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, Ph.D. Thesis, 2003. [↑54](#)
- [BFL06] Y. Boykov and G. Funka-Lea, *Graph Cuts and Efficient N-D Image Segmentation*, International Journal on Computer Vision **70** (2006), no. 2, 109–131. [↑74](#)
- [BGHM95] J. Besag, P.J. Green, D. Higdon, and K.L. Mengersen, *Bayesian computation and stochastic systems (with discussion)*, Statistical Science **10** (1995), 3–66. [↑17](#)
- [BLM04] M.H. Bharati, J.J. Liu, and J.F. MacGregor, *Image texture analysis: methods and comparisons*, Chemometrics and Intelligent Laboratory Systems **72** (2004), no. 1, 57–71. [↑22](#)
- [BMK08] S. Derin Babacan, Rafael Molina, and Aggelos K. Katsaggelos, *Parameter Estimation in TV Image Restoration Using Variational Distribution Approximation*, IEEE Transactions on Image Processing **17** (2008), 326–339. [↑9](#)
- [BMK09] S.D. Babacan, R. Molina, and A.K. Katsaggelos, *Variational Bayesian Blind Deconvolution Using a Total Variation Prior*, IEEE Transactions on Image Processing **18** (2009), no. 1, 12–26. [↑9](#), [40](#), [48](#)
- [BMK11] S. D. Babacan, R. Molina, and Aggelos K. Katsaggelos, *Variational Bayesian Super Resolution*, IEEE Transactions on Image Processing **20** (2011), 984–999. [↑9](#)
- [Boh05] T. Bohlke, *Application of the maximum entropy method in texture analysis*, Computational Materials Science **32** (2005), no. 3–4, 276–283. [↑23](#)
- [Bou13] C.A. Bouman, *A Guide to the Tools of Model Based Image Processing*, 2013. [↑24](#)

- [BPSS11] A. Beskos, F.J. Pinski, J.M. Sanz-Serna, and A.M. Stuart, *Hybrid Monte Carlo on Hilbert spaces*, Stochastic Processes and their Applications **121** (2011), no. 10, 2201–2230. [↑14](#)
- [Bro99] M. Brown, *Texture synthesis and prediction error filtering*, 1999. [↑22, 26](#)
- [BS96] C.A. Bouman and M. Shapiro, *A Multiscale Random Field Model for Bayesian Image Segmentation*, 1996. [↑24](#)
- [BTG12] T. Bui-Thanh and O. Ghattas, *A Scaled Stochastic Newton Algorithm for Markov Chain Monte Carlo Simulations* (2012). [↑14](#)
- [BWMK10] S.D. Babacan, Jingnan Wang, R. Molina, and A.K. Katsaggelos, *Bayesian Blind Deconvolution From Differently Exposed Image Pairs*, IEEE Transactions on Image Processing **19** (2010), no. 11, 2874–2888. [↑9, 40](#)
- [BZ05] A. Barbu and S. C. Zhu, *Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities.*, IEEE Transaction on Pattern Analysis and Machine Intelligence **27** (2005), no. 8, 1239–1253. [↑74](#)
- [CC08] C. Cariou and K. Chehdi, *Unsupervised texture segmentation/classification using 2-D autoregressive modeling and the stochastic expectation-maximization algorithm*, Pattern Recognition Letters **29** (2008), no. 7, 905–917. [↑24](#)
- [CC85] R. Chellappa and S. Chatterjee, *Classification of textures using Gaussian Markov random fields*, IEEE Transactions on Acoustics, Speech, and Signal Processing **33** (1985), no. 4, 959–963. [↑24](#)
- [CE07] P. Campisi and K. Egiazarian, *Blind image deconvolution: theory and applications*, CRC Press, 2007. [↑40](#)
- [CFP02] G. Celeux, F. Forbes, and N. Peyrard, *EM-based image segmentation using Potts models with external field*, Technical Report RR-4456, INRIA, 2002. [↑75](#)
- [CJ83] G.R. Cross and A.K. Jain, *Markov Random Field Texture Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **5** (1983), no. 1, 25–39. [↑24](#)
- [CJK93] T. Chang and C.-C. Jay Kuo, *Texture analysis and classification with tree-structured wavelet transform*, IEEE Transactions on Image Processing **2** (1993). [↑22](#)
- [CK85] R. Chellappa and R. Kashyap, *Texture synthesis using 2-D noncausal autoregressive models*, IEEE Transactions on Acoustics, Speech, and Signal Processing **33** (1985), no. 1, 194–203. [↑24](#)
- [CKC93] S. S. Chen, J. M. Keller, and R. M. Crownover, *On the Calculation of Fractal Features from Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993), 1087–1090. [↑24](#)
- [CNS00] P. Campisi, A. Neri, and G. Scarano, *Nonlinear Prediction in the 2D Wold Decomposition for Texture Modeling*, Proceedings of the IEEE ICIP, 2000. [↑23](#)
- [CS95] B. B. Chaudhuri and Nirupam Sarkar, *Texture Segmentation Using Fractal Dimension*, IEEE Transaction on Pattern Analysis and Machine Intelligence **17** (1995), 72–77. [↑24](#)
- [CV99] T. Chan and L. Vese, *An Active Contour Model without Edges*, International Conference on Scale-Space Theories in Computer Vision, 1999, pp. 141–151. [↑74](#)
- [DC04] H. Deng and D.A. Clausi, *Unsupervised image segmentation using a simple MRF model with a new implementation scheme*, Pattern Recognition **37** (2004), no. 12, 2323–2335. [↑75](#)
- [DV02a] M.N. Do and M. Vetterli, *Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models*, IEEE Transactions on Multimedia **4** (2002), no. 4, 517–527. [↑22](#)
- [DV02b] ———, *Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance*, IEEE Transactions on Image Processing **11** (2002), no. 2, 146–158. [↑22](#)
- [FBD10] M. Figueiredo and J.M. Bioucas-Dias, *Restoration of Poissonian Images Using Alternating Direction Optimization*, IEEE Transactions on Image Processing **19** (2010), no. 12, 3133–3145. [↑40](#)

- [FH04] P.F. Felzenszwalb and D.P. Huttenlocher, *Efficient Graph-Based Image Segmentation*, International Journal on Computer Vision **59** (2004), no. 2, 167–181. [↑74](#)
- [FX03] G. Fan and X.-G. Xia, *Wavelet-based texture analysis and synthesis using Hidden Markov models*, IEEE Transactions on Circuits Systems I **50** (2003), 106–120. [↑22](#)
- [GC11] M. Girolami and B. Calderhead, *Riemannian manifold Hamiltonian Monte Carlo (with discussion)*, Journal of the Royal Statistical Society, Series B **73** (2011), 123–214. [↑14](#), [18](#)
- [GCSR04] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, 2004. [↑11](#)
- [GD94] A.E. Gelfand and D.K. Dey, *Bayesian Model Choice: Asymptotics and Exact Calculations*, Journal of the Royal Statistical Society: Series B **56** (1994), no. 3, 501–514. [↑61](#)
- [GGGD90] D. Geman, S. Geman, C. Graffigne, and P. Dong, *Boundary Detection by Constrained Optimization*, IEEE Transaction on Pattern Analysis and Machine Intelligence **12** (1990), no. 7, 609–628. [↑75](#)
- [GGM11] B. Galerne, Y. Gousseau, and J.-M. Morel, *Random Phase Textures: Theory and Synthesis*, IEEE Transactions in Image Processing **20** (2011), no. 1, 257–267. [↑23](#)
- [Gim99] G.L. Gimel'farb, *Image Textures and Gibbs Random Fields*, Kluwer Academic Publishers, 1999. [↑24](#)
- [Gio08] J. F. Giovannelli, *Unsupervised Bayesian Convex Deconvolution Based on a Field With an Explicit Partition Function*, IEEE Transactions on Image Processing **17** (2008), no. 1, 16–26. [↑32](#), [112](#)
- [Gio11] J.-F Giovannelli, *Ising field parameter estimation from incomplete and noisy data*, Proceedings of IEEE International Conference on Image Processing, 2011, pp. 1853–1856. [↑112](#)
- [GMI13] C. Gilavert, S. Moussaoui, and J. Idier, *Rééchantillonnage gaussien en grande dimension pour les problèmes inverses*, Actes du 24<sup>e</sup> colloque gretsi, 2013. [↑83](#)
- [GR92] D. Geman and G. Reynolds, *Constrained Restoration and the Recovery of Discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 3, 367–383. [↑32](#)
- [Gra06] L. Grady, *Random Walks for Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 11, 1768–1783. [↑74](#)
- [Gre95] P.J. Green, *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*, Biometrika **82** (1995), 711–732. [↑61](#)
- [GRG96] A. Gelman, G.O. Roberts, and W.R. Gilks, *Bayesian Statistics 5*, Oxford University Press, 1996. [↑16](#), [17](#)
- [GS04] T.L. Griffiths and M. Steyvers, *Finding scientific topics*, Proceedings of the National Academy of Science USA, 2004, pp. 5228–5235. [↑59](#)
- [GSBB03] M. Galun, E. Sharon, R. Basri, and A. Brandt, *Texture segmentation by multiscale aggregation of filter responses and shape elements*, IEEE International Conference on Computer Vision, 2003, pp. 716–723. [↑75](#)
- [GUSV<sup>+</sup>09] L. Garcia Ugarrita, E. Saber, S.R. Vantaram, V. Amuso, M. Shaw, and R. Bhaskar, *Automatic Image Segmentation by Dynamic Region Growth and Multiresolution Merging*, IEEE Transactions on Image Processing **18** (2009), no. 10, 2275–2288. [↑74](#)
- [GY95] D. Geman and C. Yang, *Nonlinear Image Recovery with Half-Quadratic Regularization*, IEEE Transactions on Image Processing **4** (1995), no. 7, 932–946. [↑32](#)
- [Has70] W.K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika **57** (1970), 97–109. [↑13](#)
- [HS08] D.K. Hammond and E.P. Simoncelli, *Image Modeling and Denoising With Orientation-Adapted Gaussian Scale Mixtures*, IEEE Transactions on Image Processing **17** (2008), no. 11, 2089–2101. [↑25](#), [32](#)

- [HST01] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli **7**(2) (2001), 223–242. [↑13](#)
- [JGSF73] B. Julesz, E.N. Gilbert, L.A. Shepp, and H.L. Frisch, *Inability of humans to discriminate between visual textures that agree in second-order statistics - revisited*, Perception **2** (1973), 391–405. [↑26](#)
- [Jul62] B. Julesz, *Visual Pattern Discrimination*, IRE Transactions on Information Theory **8** (1962), no. 2, 84–92. [↑26](#)
- [Jul80] ———, *Spatial nonlinearities in the instantaneous perception of textures with identical power spectra*, Philosophical Transactions of the Royal Society of London **B290** (1980), 83–94. [↑26](#), [27](#)
- [Kas80] R.L. Kashyap, *Univariate and multivariate random field models for images*, Computer Graphics and Image Processing **12** (1980), no. 3, 257–270. [↑24](#)
- [KKS<sup>+</sup>10] K. Kayabol, E.E. Kuruouğlu, J.L. Sanz, B. Sankur, E. Salerno, and D. Herranz, *Adaptive Langevin Sampler for Separation of t-Distribution Modelled Astrophysical Maps*, IEEE Transactions on Image Processing **19** (2010). [↑14](#)
- [KR95] R.E. Kass and A.E. Raftery, *Bayes Factors*, Journal of the American Statistical Association **90** (1995), no. 430, 773–795. [↑60](#)
- [KTHD12] G. Kail, J.-Y. Tournet, F. Hlawatsch, and N. Dobigeon, *Blind Deconvolution of Sparse Pulse Sequences under a Minimum Distance Constraint: A Partially Collapsed Gibbs Sampler Method*, IEEE Transactions on Signal Processing **60** (2012), no. 6, 2727–2743. [↑62](#)
- [KW96] R.E. Kass and L. Wasserman, *The Selection of Prior Distributions by Formal Rules*, Journal of the American Statistical Association **91** (1996), no. 435, 1343–1370. [↑12](#)
- [Law80] K.I. Laws, *Rapid Texture Identification*, SPIE Conference on Image Processing for Missile Guidance, 1980, pp. 376–380. [↑22](#)
- [LB06] E. Levina and P.J. Bickel, *Texture synthesis and nonparametric resampling of Random Fields*, The Annals of Statistics **34** (2006), no. 4, 1751–1773. [↑24](#)
- [LH10] W.-L. Lee and K.-S. Hsieh, *A robust algorithm for the fractal dimension of images and its applications to the classification of natural images and ultrasonic liver images*, Signal Processing **90** (2010), 1894–1904. [↑24](#)
- [LLZ06] H. Li, G. Liu, and Z. Zhang, *A new texture generation method based on pseudo-DCT coefficients*, IEEE Transactions on Image Processing **15** (2006), no. 5, 1300–1312. [↑23](#)
- [LMS07] A. Lillo, G. Motta, and J.A. Storer, *Supervised Segmentation Based on Texture Signatures Extracted in the Frequency Domain*, Pattern recognition and image analysis, 2007, pp. 89–96. [↑74](#)
- [LP96] F. Liu and R.W. Picard, *Periodicity, directionality, and randomness: Wold features for image modeling and retrieval*, IEEE Transactions on Pattern Analysis and Machine Intelligence **7** (1996), 722–733. [↑23](#)
- [LS09] S. Lyu and E.P. Simoncelli, *Modeling Multiscale Subbands of Photographic Images with Fields of Gaussian Scale Mixtures*, IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009), no. 4, 693–706. [↑25](#), [32](#)
- [LSP05] S. Lazebnik, C. Schmid, and J. Ponce, *A Maximum Entropy Framework for Part-Based Texture and Object Recognition*, Proceedings of the International Conference on Computer Vision, 2005, pp. 832–838. [↑23](#)
- [LW06] X. Liu and D. Wang, *Image and Texture Segmentation Using Local Spectral Histograms*, IEEE Transactions on Image Processing **15** (2006), no. 10, 3066–3077. [↑75](#)
- [MBLS01] J. Malik, S. Belongie, T. Leung, and J. Shi, *Contour and Texture Analysis for Image Segmentation*, International Journal of Computer Vision **43** (2001), 7–27. [↑23](#), [74](#)
- [MH80] J. Sklansky M. Hassner, *The use of Markov Random Fields as models of texture*, Computer Graphics and Image Processing **12** (1980), no. 4, 357–370. [↑24](#)



- [Mig06] M. Mignotte, *A Segmentation-Based Regularization Term for Image Deconvolution*, IEEE Transactions on Image Processing **15** (2006), no. 7, 1973–1984. [↑75](#)
- [MMK06] R. Molina, J. Mateos, and A.K. Katsaggelos, *Blind deconvolution using a variational approach to parameter, image, and blur estimation*, IEEE Transactions on Image Processing **15** (2006), 3715–3727. [↑9, 40](#)
- [MN01] T. Maddess and Y. Nagai, *Discriminating of isotrigon textures*, Vision Research **41** (2001), no. 28, 3837–3860. [↑26](#)
- [MR12] T. Marshall and G. Roberts, *An adaptive approach to Langevin MCMC*, Statistical Computing **22** (2012), 1041–1057. [↑14](#)
- [MRR<sup>+</sup>] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, *Equations of state calculations by fast computing machines*. [↑13](#)
- [MRY<sup>+</sup>11] H. Mobahi, S. Rao, A.Y. Yang, S.S. Sastry, and Y. Ma, *Segmentation of Natural Images by Texture and Boundary Compression*, International Journal of Computer Vision **95** (2011), no. 1, 86–98. [↑74](#)
- [MS98] A. Materka and M. Strzelecki, *Texture Analysis Methods - A Review*, Institute of Electronics, Technical University of Lodz, 1998. [↑22](#)
- [MWBG12] J. Martin, L.C. Wilcox, C. Burstedde, and O. Ghattas, *A Stochastic Newton MCMC Method for Large-Scale Statistical Inverse Problems with Application to Seismic Inversion*, SIAM Journal of Scientific Computing **34** (2012), no. 3, A1460–A1487. [↑14](#)
- [Nea11] R.M. Neal, *MCMC using Hamiltonian dynamics*, Handbook of Markov chain Monte Carlo, 2011, pp. 93–112. [↑14](#)
- [NR94] M.A. Newton and A.E. Raftery, *Approximate Bayesian Inference with the Weighted Likelihood Bootstrap*, Journal of the Royal Statistical Society: Series B **56** (1994), no. 1, 3–48. [↑59, 61](#)
- [OFG12] F. Orieux, O. Feron, and J.-F. Giovannelli, *Sampling High-Dimensional Gaussian Distributions for General Linear Inverse Problems*, IEEE Signal Processing Letters **19** (2012), no. 5, 251–254. [↑83](#)
- [OGR10] F. Orieux, J.F. Giovannelli, and T. Rodet, *Bayesian estimation of regularization and Point Spread Function parameters for Wiener-Hunt deconvolution*, Journal of Optical Society of America A **27** (2010), no. 7, 1593–1607. [↑40](#)
- [Pak99] A.G. Pakes, *On the convergence of moments of geometric and harmonic means*, Statistica Neerlandica **53** (1999), no. 1, 96–110. [↑59](#)
- [PDBT13] M. Pereyra, N. Dobigeon, H. Batatia, and J.-Y. Tournier, *Estimating the Granularity Coefficient of a Potts-Markov Random Field Within a Markov Chain Monte Carlo Algorithm*, IEEE Transactions on Image Processing **22** (2013), no. 6, 2385–2397. [↑75](#)
- [PDH12] S.U. Park, N. Dobigeon, and A.O. Hero, IEEE Transactions on Image Processing **21** (2012), no. 9, 3838–3849. [↑40](#)
- [PDH14] ———, *Variational semi-blind sparse deconvolution with orthogonal kernel bases and its application to MRFM*, Signal Processing **94** (2014), 386–400. [↑40](#)
- [P.W] P. Whittle, *The Analysis of Multiple Stationary Time Series*, Journal of the Royal Statistical Society. Series B (Methodological). [↑26](#)
- [PS00] J. Portilla and E.P. Simoncelli, *A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients*, International Journal of Computer Vision **40** (2000), no. 1, 49–71. [↑22](#)
- [PSWS03] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli, *Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain*, IEEE Transactions on Image Processing **12** (2003), 1338–1351. [↑25, 32](#)
- [QM02] Y. Qi and T.P. Minka, *Hessian-based Markov Chain Monte-Carlo Algorithms*, Workshop on Monte Carlo Methods (2002). [↑14, 17, 19](#)

- [Raf95] A.E. Raftery, *Hypothesis Testing and Model Selection via Posterior Simulation*, Practical Markov Chain Monte Carlo, 1995. [↑59](#), [60](#)
- [RNSK07] A.E. Raftery, M.A. Newton, J.M. Satagopan, and P.N. Krivitsky, *Estimating the integrated likelihood via posterior simulation using the harmonic mean identity*, Bayesian Statistics, 2007, pp. 1–45. [↑59](#)
- [Ros11] J. Rosenthal, *Optimal proposal distributions and adaptive MCMC*, Handbook of Markov chain Monte Carlo, 2011, pp. 93–112. [↑13](#)
- [RS03] G. Roberts and O. Stramer, *Langevin Diffusions and Metropolis-Hastings Algorithms*, Methodology and Computing in Applied Probability **4** (2003), 337–358. [↑14](#)
- [RW09] C.P. Robert and D. Wraith, *Computational methods for Bayesian model choice*, 2009. [↑59](#), [61](#)
- [SBCL02] S. D. Spiegelhalter, N. G. Best, B. P. Carlin, and A. V. D. Linde, *Bayesian measures of model complexity and fit*, Journal of the Royal Statistical Society: Series B **64** (2002), no. 4, 583–639. [↑54](#)
- [Sch78] G. Schwarz, *Estimating the Dimension of a Model*, The Annals of Statistics **6** (1978), no. 2, 461–464. [↑54](#)
- [SFP96] R. Sriram, J.-M. Francos, and W.A. Pearlman, *Texture coding using a Wold decomposition model*, IEEE Transactions on Image Processing **5** (1996), 1382–1386. [↑23](#)
- [SG07] A.K. Sinop and L. Grady, *A Seeded Image Segmentation Framework Unifying Graph Cuts And Random Walker Which Yields A New Algorithm*, IEEE International Conference on Computer Vision, 2007, pp. 1–8. [↑74](#)
- [Sim05] Eero P. Simoncelli, *Statistical Modeling of Photographic Images*, Handbook of Video and Image Processing, 2005. [↑25](#), [32](#)
- [Ski06] J. Skilling, *Nested Sampling for General Bayesian Computation*, Bayesian Analysis **1** (2006), no. 4, 833–860. [↑58](#)
- [SS01a] M. Sharma and S. Singh, *Evaluation of texture methods for image analysis*, Intelligent Information Systems Conference, 2001, pp. 117–121. [↑22](#)
- [SS01b] G. Stockman and L.G. Shapiro, *Computer Vision*, 1st ed., Prentice Hall PTR, 2001. [↑21](#)
- [SS08] G.N. Srinivasan and G. Shobha, *Statistical Texture Analysis*, Proceedings of World Academy of Science, Engineering and Technology, 2008, pp. 1264–1269. [↑22](#)
- [STNR05] Y. Stitou, F. Turcu, M. Najim, and L. Redouane, *3-D Texture Characterization Based on Wold Decomposition and Higher Order Statistics*, Proceedings of IEEE ICASSP, 2005, pp. 165–168. [↑23](#)
- [Stu10] A.M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numerica **19** (2010), 451–559. [↑11](#)
- [TA09] S. Todorovic and N. Ahuja, *Texel-based texture segmentation*, IEEE International Conference on Computer Vision, 2009, pp. 841–848. [↑75](#)
- [Tie94] L. Tierney, *Markov chains for exploring posterior distributions*, Annals of Statistics **22** (1994), 1701–1762. [↑13](#)
- [TJ98] M. Tuceryan and A.K. Jain, *Texture Analysis*, 1998. [↑22](#)
- [TLG07] D.G. Tzikas, A.C.. Likas, and N.P. Galatsanos, *Variational Bayesian Blind Image Deconvolution with Student-T Priors*, IEEE International Conference on Image Processing, 2007, pp. I –109–I –112. [↑40](#)
- [TLG09] D.G. Tzikas, A.C. Likas, and N.P. Galatsanos, *Variational Bayesian sparse kernel-based blind image deconvolution with Student’s-T priors*, IEEE Transactions on Image Processing **18** (2009), no. 4, 753–764. [↑40](#)
- [Tuc94] M. Tuceryan, *Moment-based texture segmentation.*, Pattern Recognition Letters **15** (1994), no. 7, 659–668. [↑74](#)

- [TZS01] Z. Tu, S.-C. Zhu, and H.-Y. Shum, *Image segmentation by data driven Markov chain Monte Carlo*, IEEE International Conference on Computer Vision, 2001, pp. 131–138. [↑75](#)
- [VGB11] C. Vacar, J.-F. Giovannelli, and Y. Berthoumieu, *Langevin and Hessian with Fisher approximation stochastic sampling for parameter estimation of structured covariance*, Proceedings of the IEEE ICASSP, 2011, pp. 3964–3967. [↑18, 27, 57, 106](#)
- [VGB14] ———, *Bayesian Texture and Instrument Parameter Estimation From Blurred and Noisy Images Using MCMC*, IEEE Signal Processing Letters **21** (2014), no. 6, 707–711. [↑38, 71](#)
- [VGB] ———, *Bayesian Texture Model Choice from Indirect Observations using Fast Sampling*. [↑57, 59](#)
- [VGR12] C. Vacar, J.-F. Giovannelli, and A.-M. Roman, *Bayesian Texture Model Selection using Harmonic Mean*, Proceedings of IEEE ICIP, 2012. [↑27, 59, 63](#)
- [Vic94] J.D. Victor, *Images, statistics, and textures: implications of triple correlation uniqueness for texture statistics and the Julesz conjecture: comment*, JOSA A **11** (1994), 1680–1684. [↑26](#)
- [VU08] C. Vonesch and M. Unser, *A fast thresholded Landweber algorithm for wavelet regularized multidimensional deconvolution*, IEEE Transactions Image Processing (2008), 539–549. [↑40](#)
- [Wal06] H.M. Wallach, *Topic modeling: beyond bag-of-words*, Proceedings of the ICML, 2006, pp. 977–984. [↑59](#)
- [WHMM06] L. Wolf, X. Huang, I. Martin, and D. Metaxas, *Patch-based texture edges and segmentation*, In European Conference on Computer Vision, 2006. [↑74](#)
- [Win06] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction (Stochastic Modelling and Applied Probability)*, Springer-Verlag New York, Inc., 2006. [↑80](#)
- [WL00] L.-Y. Wei and M. Levoy, *Fast texture synthesis using tree-structured vector quantization*, Proceedings of the 27th Conference on Computer graphics and interactive techniques, 2000, pp. 479–488. [↑23](#)
- [WSW01] M.J. Wainwright, E.P. Simoncelli, and A.S. Willsky, *Random Cascades on Wavelet Trees and Their Use in Analyzing and Modeling Natural Images*, Applied and Computational Harmonic Analysis **11** (2001), 89–123. [↑25, 32](#)
- [XFZ06] Y. Xia, D.D. Feng, and R. Zhao, *Morphology-based multifractal estimation for texture segmentation*, IEEE Transactions on Image Processing **15** (2006), no. 3, 614–623. [↑24](#)
- [ZFS01] F. Zhou, J.-F. Feng, and Q.-Y. Shi, *Texture feature based on local Fourier transform*, Proceedings of the International Conference on Image Processing, 2001, pp. 610–613. [↑23](#)
- [ZS11] Y. Zhang and C. Sutton, *Quasi-Newton methods for Markov chain Monte-Carlo*, Advances in Neural Information Processing Systems **24** (2011). [↑14](#)
- [ZWM97] S.C. Zhu, Y.N. Wu, and D. Mumford, *Minimax Entropy Principle and Its Application to Texture Modeling*, Neural Computation **9** (1997), 1627–1660. [↑23](#)
- [ZWM98] S.C. Zhu, Y. Wu, and D. Mumford, *Filters, Random fields And Maximum Entropy (FRAME) – Towards a Unified Theory for Texture Modeling*, International Journal of Computer Vision **27** (1998), no. 2, 1–20. [↑23](#)
- [ZZC10] J. Zhang, J. Zheng, and J. Cai, *A diffusion approach to seeded image segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2125–2132. [↑74](#)